Universal Agent 7.2.x

Installation, Upgrade, and Applying Maintenance

© 2022 by Stonebranch, Inc.All Rights Reserved.

1. Universal Agent 7.2.x Installation Information . . . 4
1.1 Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance . . . 5
1.1.1 Installation, Upgrade, and Applying Maintenance Overview . . . 6
1.1.2 Installing Universal Agent . . . 7
1.1.3 Installing Universal Agent - Universal Connector for SAP Prerequisite . . . 11
1.1.4 z/OS Installation . . . 14
1.1.4.1 z/OS Installation - Installation Package . . . 16
1.1.4.2 z/OS Installation - Installation Requirements . . . 17
1.1.4.3 z/OS Installation - Distribution File . . . 18
1.1.4.4 z/OS Installation - Transferring Installation Files to z/OS . . . 22
1.1.4.5 z/OS Installation - SMPE Installation . . . 32
1.1.4.5.1 z/OS Installation - New Install, New CSI . . . 34
1.1.4.5.2 z/OS Installation - Universal Agent 7.1.x Upgrade, Existing CSI . . . 36
1.1.4.5.3 z/OS Installation - Universal Agent 7.0.x Upgrade, Existing CSI . . . 37
1.1.4.5.4 z/OS Installation - Universal Agent 6.9.x Upgrade, Existing CSI . . . 38
1.1.4.5.5 z/OS Installation - Universal Agent 6.8.x Upgrade, Existing CSI . . . 39
1.1.4.5.6 z/OS Installation - Universal Agent 6.7.x Upgrade, Existing CSI . . . 40
1.1.4.5.7 z/OS Installation - Universal Agent 6.6.x Upgrade, Existing CSI . . . 41
1.1.4.5.8 z/OS Installation - Universal Agent 6.5.x Upgrade, Existing CSI . . . 42
1.1.4.5.9 z/OS Installation - Universal Agent 6.4.x Upgrade, Existing CSI . . . 43
1.1.4.5.10 z/OS Installation - Universal Agent 6.3.x Upgrade, Existing CSI . . . 44
1.1.4.5.11 z/OS Installation - Universal Agent 6.2.x Upgrade, Existing CSI . . . 45
1.1.4.5.12 z/OS Installation - Universal Agent 5.2.0 Upgrade, Existing CSI . . . 46
1.1.4.5.13 z/OS Installation - Workload Automation 5.1.0 Upgrade, Existing CSI . . . 47
1.1.4.5.14 z/OS Installation - Stonebranch Solutions 4.x Upgrade, Existing CSI . . . 48
1.1.4.5.15 z/OS Installation - Universal Products 3.2.0 Upgrade, Existing CSI . . . 49
1.1.4.6 z/OS Installation - Configuration . . . 50
1.1.4.6.1 z/OS Configuration - Started Tasks . . . 51
1.1.4.6.2 z/OS Configuration - Load Library . . . 57
1.1.4.6.3 z/OS Configuration - SMF Exits . . . 60
1.1.4.6.4 z/OS Configuration - JES SYSOUT Processing . . . 64
1.1.4.6.5 Universal Agent Database Configuration . . . 66
1.1.4.7 z/OS Installation - Customization . . . 69
1.1.4.8 z/OS Installation - Cumulative PTF Maintenance . . . 75
1.1.4.9 z/OS Installation - Sysplex . . . 80
1.1.4.10 z/OS Installation - Time Zone Environment Variable . . . 87
1.1.4.11 z/OS Installation - TCP/IP Configuration . . . 89
1.1.4.12 z/OS Installation - SAP RFC DLL . . . 91
1.1.4.13 z/OS Installation - Configuration of z/OS System SSL . . . 93
1.1.4.13.1 Integrated Cryptographic Service Facility (ICSF) . . . 94
1.1.4.13.2 Universal Broker Digital Certificate (RACF) Set-up . . . 95
1.1.4.14 z/OS Installation - Configuration of Security . . . 97
1.1.4.14.1 Configuration of Security - RACF Class . . . 98
1.1.4.14.2 Configuration of Security - ACF2 Class . . . 100
1.1.4.14.3 Configuration of Security - Universal Command Security . . . 101
1.1.4.14.4 Configuration of Security - Universal Control Security . . . 104
1.1.4.14.5 Configuration of Security - Universal Event Monitor Security . . . 105
1.1.4.15 z/OS Installation - Performance Guidelines . . . 106
1.1.4.16 z/OS Installation - Converting STC User Profiles to a Non-Zero UID . . . 108
1.1.4.17 z/OS Installation - Data Set Inventory . . . 113
1.1.4.18 z/OS Installation - Licensing . . . 115
1.1.5 z/OS USS Installation . . . 117
1.1.5.1 z/OS USS Installation - Installation Package . . . 118
1.1.5.2 z/OS USS Installation - Installation Requirements . . . 119
1.1.5.3 z/OS USS Installation - Distribution File . . . 121
1.1.5.4 z/OS USS Installation - SMPE Installation . . . 122
1.1.5.4.1 z/OS USS Installation - New Install, New CSI . . . 123
1.1.5.4.2 z/OS USS Installation - 7.1.x Upgrade, Existing CSI . . . 124
1.1.5.4.3 z/OS USS Installation - 7.0.x Upgrade, Existing CSI . . . 125
1.1.5.4.4 z/OS USS Installation - 6.9.x Upgrade, Existing CSI . . . 126
1.1.5.4.5 z/OS USS Installation - 6.8.x Upgrade, Existing CSI . . . 127
1.1.5.4.6 z/OS USS Installation - 6.7.x Upgrade, Existing CSI . . . 128
1.1.5.4.7 z/OS USS Installation - 6.6.x Upgrade, Existing CSI . . . 129
1.1.5.4.8 z/OS USS Installation - 6.5.x Upgrade, Existing CSI . . . 130
1.1.5.4.9 z/OS USS Installation - 6.4.x Upgrade, Existing CSI . . . 131
1.1.5.4.10 z/OS USS Installation - 6.3.x Upgrade, Existing CSI . . . 132
1.1.5.4.11 z/OS USS Installation - 6.2.x Upgrade, Existing CSI . . . 133
1.1.5.4.12 z/OS USS Installation - 5.2.0 Upgrade, Existing CSI . . . 134
1.1.5.4.13 z/OS USS Installation - 5.1.0 Upgrade, Existing CSI . . . 135
1.1.5.4.14 z/OS USS Installation - 4.x Upgrade, Existing CSI . . . 136
1.1.5.4.15 z/OS USS Installation - 3.2.0 Upgrade, Existing CSI . . . 137
1.1.5.5 z/OS USS Installation - Customization . . . 138
1.1.5.6 z/OS USS Installation - Data Set Inventory . . . 140
1.1.5.7 z/OS USS Installation - Licensing . . . 141
1.1.6 Windows Installation . . . 143
1.1.6.1 Universal Agent for Windows System Installation . . . 149
1.1.6.1.1 Universal Agent for Windows - System Installation Package . . . 150
1.1.6.1.2 Universal Agent for Windows - System Installation Requirements . . . 151
1.1.6.1.3 Universal Agent for Windows - System Installation Procedures . . . 154

1.1.6.1.4 Universal Agent for Windows - System Installation File Inventory Lists . . . 184
1.1.6.2 Universal Agent for Windows User Mode Installation . . . 194
1.1.6.2.1 Universal Agent for Windows - User Mode Installation Package . . . 195
1.1.6.2.2 Universal Agent for Windows - User Mode Installation Requirements . . . 197
1.1.6.2.3 Universal Agent for Windows - User Mode Installation Procedures . . . 200
1.1.6.2.4 Universal Agent for Windows - User Mode Installation File Inventory Lists . . . 215
1.1.6.3 Universal Enterprise Controller for Windows Installation . . . 223
1.1.6.3.1 Universal Enterprise Controller for Windows - Installation Package . . . 224
1.1.6.3.2 Universal Enterprise Controller for Windows - Installation Requirements . . . 225
1.1.6.3.3 Universal Enterprise Controller for Windows - Installation Procedures . . . 226
1.1.6.3.4 Universal Enterprise Controller for Windows - 64-Bit Windows Editions . . . 247
1.1.6.3.5 Universal Enterprise Controller for Windows - Database Configuration . . . 249
1.1.6.3.6 Universal Enterprise Controller for Windows - File Inventory Lists . . . 250
1.1.6.4 UEC Client Applications Installation . . . 252
1.1.6.4.1 UEC Client Applications - Installation Package . . . 253
1.1.6.4.2 UEC Client Applications - Installation Requirements . . . 254
1.1.6.4.3 UEC Client Applications - Installation Procedures . . . 255
1.1.6.4.4 UEC Client Applications - 64-Bit Windows Editions . . . 279
1.1.6.4.5 UEC Client Applications - File Inventory Lists . . . 280
1.1.6.5 Universal Agent for SOA for Windows Installation . . . 282
1.1.6.5.1 Universal Agent for SOA for Windows - Installation Package . . . 283
1.1.6.5.2 Universal Agent for SOA for Windows - Installation Requirements . . . 284
1.1.6.5.3 Universal Agent for SOA for Windows - Pre-Installation - Upgrade Backups . . . 285
1.1.6.5.4 Universal Agent for SOA for Windows - Installation Procedures . . . 286
1.1.6.5.5 Universal Agent for SOA for Windows - Configuring and Starting UAC Server . . . 301
1.1.6.5.6 Universal Agent for SOA for Windows - 64-Bit Windows Editions . . . 302
1.1.6.5.7 Universal Agent for SOA for Windows - File Inventory Lists . . . 303
1.1.6.6 Windows Installation - Licensing . . . 306
1.1.7 UNIX Installation . . . 309
1.1.7.1 Universal Agent for UNIX Installation . . . 311
1.1.7.1.1 Universal Agent for UNIX - Installation Package . . . 312
1.1.7.1.2 Universal Agent for UNIX - Installation Requirements . . . 313
1.1.7.1.3 Universal Agent for UNIX - Distribution File . . . 317
1.1.7.1.4 Universal Agent for UNIX - Installation Procedures . . . 318
1.1.7.1.5 Universal Agent for UNIX - Customization . . . 359
1.1.7.1.6 Universal Agent for UNIX - File Inventory Lists . . . 363
1.1.7.2 Universal Agent for SOA for UNIX Installation . . . 373
1.1.7.2.1 Universal Agent for SOA for UNIX - Installation Package . . . 374
1.1.7.2.2 Universal Agent for SOA for UNIX - Installation Requirements . . . 375
1.1.7.2.3 Universal Agent for SOA for UNIX - Pre-Installation-Upgrade Backups . . . 376
1.1.7.2.4 Universal Agent for SOA for UNIX - Distribution File . . . 377
1.1.7.2.5 Universal Agent for SOA for UNIX - Deployment Options . . . 378
1.1.7.2.6 Universal Agent for SOA for UNIX - Installation Procedures . . . 381
1.1.7.2.7 Universal Agent for SOA for UNIX - File Inventory Lists . . . 385
1.1.7.3 UNIX Installation - Licensing . . . 389
1.1.8 IBM i Installation . . . 391
1.1.8.1 IBM i Installation - Installation Package . . . 393
1.1.8.2 IBM i Installation - Installation Requirements . . . 394
1.1.8.3 IBM i Installation - Distribution File . . . 397
1.1.8.4 IBM i Installation - Transferring to IBM i . . . 399
1.1.8.5 IBM i Installation - Installation Procedures . . . 401
1.1.8.5.1 IBM i Installation - New Installation (Default) . . . 402
1.1.8.5.2 IBM i Installation - New Installation (Custom) . . . 403
1.1.8.5.3 IBM i Installation - Re-Installation of Same Release . . . 404
1.1.8.5.4 IBM i Installation - Upgrade Installation to New Release . . . 405
1.1.8.5.5 IBM i Installation - Propagating New Release to Additional Systems . . . 407
1.1.8.5.6 IBM i Installation - Upgrade Installation for Maintenance Release . . . 409
1.1.8.5.7 IBM i Installation - UCHGRLS (Change Release Tag) Program . . . 410
1.1.8.5.8 IBM i Installation - Product Removal . . . 412
1.1.8.6 IBM i Installation - Customization . . . 417
1.1.8.7 IBM i Installation - Object Inventory Lists . . . 421
1.1.8.8 IBM i Installation - Licensing . . . 425
1.1.9 Verifying Universal Agent Installation . . . 427
1.1.10 Upgrading Universal Agent . . . 428
1.1.11 Applying Maintenance to Universal Agent . . . 431
1.1.12 Licenses for Third-Party Libraries . . . 432
1.2 Universal Agent 7.2.x Installation Requirements and Summary . . . 507
1.2.1 Installation Requirements and Summary - z/OS . . . 509
1.2.2 Installation Requirements and Summary - Windows . . . 511
1.2.3 Installation Requirements and Summary - UNIX . . . 518
1.2.4 Installation Requirements and Summary - IBM i . . . 522
1.3 Universal Agent 7.2.x Installation Quick Start Guides . . . 526
1.3.1 Universal Agent 7.2.x for z/OS Quick Start Guide . . . 527
1.3.2 Universal Agent 7.2.x for Windows Quick Start Guide . . . 531
1.3.3 Universal Agent 7.2.x for UNIX Quick Start Guide . . . 536

Universal Agent 7.2.x Installation Information

Universal Agent 7.2.x Installation Information

The following information is provided for the installation of Universal Agent 7.2.x:

Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance	Provides platform-specific detailed information on the requirements, installation (installing, upgrading, and applying maintenance), and licensing of Universal Agent 7.2.x.
Universal Agent 7.2.x Installation Requirements and Summary	Provides platform-specific installation requirements and summary information on the installation of Universal Agent 7.2.x.
Universal Agent 7.2.x Installation Quick Start Guides	Provides platform-specific information on requirements, installation, licensing, and start-up of Universal Agent.

Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance

Overview

Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance provides information on installing, upgrading, and applying maintenance to all Universal Agent packages on all supported operating systems.

(Read the Installation, Upgrade, and Applying Maintenance Overview page to understand the differences between these three procedures.)

Detailed Information

The following pages provide detailed information for installation and administration:

- Installation, Upgrade, and Applying Maintenance Overview
- Installing Universal Agent
- Installing Universal Agent - Universal Connector for SAP Prerequisite
- z/OS Installation
- z/OS USS Installation
- Windows Installation
- UNIX Installation
- IBM i Installation
- Verifying Universal Agent Installation
- Upgrading Universal Agent
- Applying Maintenance to Universal Agent
- Licenses for Third-Party Libraries

Installation, Upgrade, and Applying Maintenance Overview

- Installation, Upgrade, and Applying Maintenance
  - Installation
  - Upgrade
  - Applying Maintenance

Installation, Upgrade, and Applying Maintenance

There are separate procedures for installing, upgrading, and applying maintenance to Universal Agent.

Installation

Installation refers to the installation of an Agent on a machine with any supported platform that does not already contain an installed Agent.

If you are installing Universal Agent for the first time, see Installing Universal Agent for instructions.

Upgrade

Upgrading refers to the increase of a currently installed Version, Release, or Modification level (see Versioning) of an Agent on a machine to a later Version, Release, or Modification level of that Agent (for example, upgrading Universal Agent 6.2.0.1 to Universal Agent 7.2.0.0).

If you are upgrading from a previous version of Universal Agent, see Upgrading Universal Agent for instructions.

Applying Maintenance

Applying maintenance refers to the increase of a currently installed Maintenance level (see Versioning) of an Agent on a machine to a later Maintenance level of that Agent (for example, applying maintenance to Universal Agent 7.2.0.0 to increase its maintenance level to 7.2.0.1).

If you are applying maintenance to your version of Universal Agent, see Applying Maintenance to Universal Agent.

Installing Universal Agent

- Overview
- Installation Requirements
  - Platform Requirements
  - Space Requirements
  - Network Requirements
  - Additional Requirements
- Ports Configuration
- Product Distribution File Download
- Product Licensing
- Additional Information

Overview

These pages provide information and instructions for the installation of Universal Agent, which differ for each supported operating system.

Note

- If you are upgrading from a previous version of the Agent - for example, from Agent 6.2.0.1 to 7.2.0.0 - see Upgrading Universal Agent for instructions.
- If you are applying maintenance to the Agent - for example, applying maintenance to Agent 7.2.0.0 to increase its maintenance level to Agent 7.2.0.1 - see Applying Maintenance to Universal Agent for instructions.

The following table identifies the different operating systems on which an Agent can be installed, and the Agent packages that can be installed on each system.

Operating System	Universal Agent	Universal Enterprise Controller	UEC Client Applications	Universal Agent for SOA
z/OS [1]	✓			
z/OS USS	✓			
Windows	✓	✓	✓	✓
UNIX - AIX	✓			
UNIX - HP-UX	✓			
UNIX - Solaris	✓			
UNIX - Linux	✓			✓
IBM i [2]	✓			

[1] Universal Agent for z/OS contains Universal Enterprise Controller.
[2] Workload Automation 5.1.1 is installed on IBM i.

For each Universal Agent installation package, the following information is provided:

- Installation package components
- Product compatibility
- Installation requirements
- Installation upgrades
- System upgrades
- Distribution files
- Installation procedures
- Product customization
  - Configuration
  - Licensing
- File inventory lists

Note

You can install Agents before, during, or after installation of Universal Controller.

Installation Requirements

Platform-specific installation requirements, including system requirements, are located on the Installation Requirements page specific to that platform:

- z/OS Installation Requirements
- Windows System Installation Requirements
- Windows User Mode Installation Requirements
- UNIX Installation Requirements
- IBM i Installation Requirements

Requirements relevant to all platforms are shown below.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support page to make sure that your platform is supported before performing an installation.

Space Requirements

Space requirements for Universal Agent are driven largely by logging. A minimum 500MB of space is recommended. Each Agent contains a Universal Broker log and an Agent log.

The Universal Broker LOG_FILE_GENERATIONS configuration option lets you specify how many log files to save in the Universal Broker log directory. The default is 5.

The Agent log rolls over at midnight (12:00 a.m.) or whenever the Agent is restarted. The Log File Retention Period in Days Universal Controller system property lets you specify the number of days that an Agent log file (in addition to a Controller log file) is retained before it is purged. The default is 5 days.

Network Requirements

Agent components run on z/OS, Windows, UNIX, and IBM i operating systems. These systems must be connected with a network that supports TCP/IP. For example, the Universal Command Manager running on a z/OS system must be able to establish a TCP/IP socket connection with the Universal Command Server running on a UNIX or Windows system.

TCP/IP provides a set of commands to verify network connectivity between systems. For example, the ping command can determine if an IP connection is possible between two systems. However, the ping command may not work in all network environments. In addition, the ping command may work, but a firewall may deny all other connection attempts between the two systems. Check with your local network administrator to determine the capabilities and topology of your network.

Universal Agent offers configuration parameters that can facilitate connections through network firewalls. Due to the large variety of firewall configurations, all possibilities cannot be discussed in this document. Check with your local network administrator to determine if a firewall is between the computer systems involved.

Additional Requirements

In order to install any of these Agent components, you must be able to write to the directory from which the installation is launched.

Ports Configuration

Universal Agent is composed of several components that utilize the TCP/IP network for component communications. TCP/IP routes must be available between the distributed components and firewall rules open, if necessary.

The primary components using TCP/IP are Universal Broker, Universal Managers, and Universal Automation Center Agent (UAG).

Universal Broker	Executes on all servers on which workload management services are required. The Universal Broker accepts TCP socket connections on port 7887 from remote clients such as Universal Command Manager and Universal Data Mover Manager. A server running Universal Agent will need to be able to accept TCP connections on port 7887 in order to provide such services.
Universal Manager	Components include Universal Command Manager, Universal Data Mover Manager, and others. Manager components initiate TCP socket connections to remote Universal Brokers. If Universal Manager services are required, the Managers will need to be able to open TCP connections to remote Universal Brokers on port 7887.
UAG	Initiates a TCP socket connection to the Universal Message Service (OMS) deployed on a remote, centrally located server. OMS accepts TCP connections on port 7878. UAG must be able to open a TCP connection to the remote OMS server on port 7878.

The port numbers above are default port numbers. They can be changed in the respective component configuration, if necessary.

Product Distribution File Download

To install, update, or apply maintenance to an Agent package, you first must download the corresponding product distribution file for your specific operating system version and hardware platform.

All Agent distribution files are available for download from the Stonebranch Customer Portal. A customer user name and password - provided by Stonebranch, Inc. - are required to access the Customer Portal.

The operating system-specific pages of this installation guide identify the product distribution files to be downloaded for the different Agent packages.

Note

Agent installation packages for z/OS, Windows, and UNIX contain the Universal Controller Command Line Interface (CLI).

Agent installation packages for Windows and UNIX contain the Universal Message Service (OMS).

To download a product distribution file:

Step 1	Log in to the Customer Portal.
Step 2	Click the **Software** link.
Step 3	Click the **Universal Agent** link
Step 4	Click the Universal Agent package link appropriate for your platform.
Step 5	Click **Save File** and browse to your save location. You can then install the software.

Product Licensing

Each Agent installation package includes one or more components that must be licensed before they can be used. Licenses are provided by your Stonebranch, Inc. account representative.

Exception

UEC Client Applications, a separate, optional installation package of Universal Agent for Windows, does not contain any components that must be licensed.

The following table identifies all licensable components in the Agent package for each supported platform. The platform name is a link to detailed information about licensing the Agent components for that platform.

Component	z/OS	z/OS USS	Windows	UNIX	IBM i
Universal Command Manager	✓	✓	✓	✓	✓
Universal Data Mover Manager	✓	✓	✓	✓	✓
Universal Connector	✓		✓	✓	
Universal Event Monitor Server			✓	✓	
Universal Enterprise Controller	✓		✓		
Universal Application Container Server			✓	✓	

Note

Universal Enterprise Controller is a separate, optional installation package of Universal Agent for Windows.

Universal Application Container Server is a component of Universal Agent for SOA, a separate, optional installation package of Universal Agent for Windows and Universal Agent for UNIX.

Additional Information

In addition to this Installation Guide, Stonebranch, Inc. provides the following information:

Universal Agent 7.2.x Installation Requirements and Summary identifies network requirements for all Agent installations and, for each operating system, the following information:

- System requirements
- Installation summary
- Customization

Universal Agent 7.2.x Installation Quick Start Guides provide the following information:

- System Requirements
- Downloading Universal Agent 7.0.x Distribution File
- Installing Universal Agent 7.0.x
- Licensing your Universal Agent 7.0.x Components
- Encrypting User ID and Password for Job Execution
- Executing a Job
- Running a System Query

Note

There is a separate quick start guide for each Universal Agent and Universal Data Mover supported platform.

Installing Universal Agent - Universal Connector for SAP Prerequisite

- Introduction
- Download
- Install
- Windows Install
- UNIX Install
  - Preparation of the RFC libraries for Linux
  - Preparation of the RFC libraries for SUSE Linux
  - File Content of profile.local

Introduction

Universal Connector for SAP requires the SAP NW (NetWeaver) RFC libraries to run. Before executing Universal Connector for SAP, you must first install the appropriate SAP NW RFC Libraries for the platform on which the connector agent is installed.

Download

To acquire the SAP libraries, access the SAP Service Marketplace and download the appropriate package for your platform:

https://launchpad.support.sap.com/#/softwarecenter/template/products/%20_APP=00200682500000001943&_EVENT=DISPHIER&HEADER=Y&FUNCTIONBAR=N&EVENT=TREE&NE=NAVIGATE&ENR=01200314690100002214&V=MAINT&TA=ACTUAL&PAGE=SEARCH/SAP%20NW%20RFC%20SDK

Note

In order to extract the downloaded SAP NW RFC package, you must use the SAPCAR executable. That utility program can be found here:

https://support.sap.com/software/patches/a-z-index.html

In order to download products from the SAP Service Marketplace you will need an SAP S-User ID with the "Download Software" authorization, which you can request via your company's SAP user administrator.

Install

Once the appropriate package has been downloaded, it can be extracted using the SAPCAR utility. The command line to extract an SAP NW RFC package is basically the same on all platforms. The only difference is the name of the SAP NW RFC package. The following is a sample command line to extract the SAP NW RFC package and its resulting output:

```
C:\tmp\sap>sapcar -xvf NWRFC_35-20004568.SAR
SAPCAR: processing archive NWRFC_35-20004568.SAR (version 2.01)
x nwrfcsdk
x nwrfcsdk/bin
x nwrfcsdk/bin/rfcexec.exe
x nwrfcsdk/bin/startrfc.exe
x nwrfcsdk/demo
x nwrfcsdk/demo/companyClient.c
x nwrfcsdk/demo/readme.txt
x nwrfcsdk/demo/rfcexec.cpp
x nwrfcsdk/demo/rfcexec.h
x nwrfcsdk/demo/sapnwrfc.ini
x nwrfcsdk/demo/sflightClient.c
x nwrfcsdk/demo/sso2sample.c
x nwrfcsdk/demo/startrfc.cpp
x nwrfcsdk/demo/startrfc.h
x nwrfcsdk/demo/stfcDeepTableServer.c
x nwrfcsdk/doc
x nwrfcsdk/include
x nwrfcsdk/include/sapdecf.h
x nwrfcsdk/include/sapnwrfc.h
x nwrfcsdk/include/sapuc.h
x nwrfcsdk/include/sapucx.h
x nwrfcsdk/lib
x nwrfcsdk/lib/icudt34.dll
x nwrfcsdk/lib/icuin34.dll
x nwrfcsdk/lib/icuuc34.dll
x nwrfcsdk/lib/libicudecnumber.dll
x nwrfcsdk/lib/libsapucum.dll
x nwrfcsdk/lib/libsapucum.lib
x nwrfcsdk/lib/sapdecfICUlib.lib
x nwrfcsdk/lib/sapnwrfc.dll
x nwrfcsdk/lib/sapnwrfc.lib
x SIGNATURE.SMF
SAPCAR: 32 file(s) extracted
```

Windows Install

From the extracted SAP NW RFC library, the dll files located in `nwrfcsdk/lib/` must be placed in a directory path that is listed in the environment variable `Path`.

This can be done by adjusting the `Path` environment variable to include the `nwrfcsdk/lib/` path or by moving the dll files from the SAP NW RFC SDK archive to a location that already exists in the `Path`.

Note

It is sufficient to place the SAP NW RFC `*.dll` files in the Universal Connector's `bin` directory. That path for a typical Windows install is `C:\Program Files\Universal\USap\bin`.

NetWeaver RFC LIbrary Dependencies

The runtime dependencies form the SAP NetWeaver RFC libraries will likely differ from Universal Agent dependencies. One known issue is that the NW RFC 750 libraries require a different version of the Microsoft Visual C++ runtime libraries than the required for the Universal Agent. If the runtime dependencies for those libraries are not available on your Agent system, attempts to run USAP will fail. To review runtime requirements for the NetWeaver RFC SDK version 750, refer to the information on this page.

UNIX Install

From the extracted SAP NW RFC library, the shared library files located in `nwrfcsdk/lib/` must be placed in a directory path that is listed in the environment variable `LD_LIBRARY_PATH`.

This can be done by adjusting the `LD_LIBRARY_PATH` environment variable to include the `nwrfcsdk/lib/` path or by moving the shared library files from the SAP NW RFC SDK archive to a location that already exists in `LD_LIBRARY_PATH`.

Preparation of the RFC libraries for Linux

Step 1	Download the appropriate tar file for the target platform: `SAP_NW_RFC_Lin_x64.tar` or `SAP_NW_RFC_Lin_x86.tar` (64-bit and 32-bit respectively)

Step 2	Create the `/usr/local/lib/sap/` directory for the SAP libraries to reside in. `md /usr/local/lib/sap`
Step 3	Copy the RFC tar file into the directory (from Step 2) and extract the contents: `$tar -xvf SAP_NW_RFC_Lin_x64.tar`
Step 4	Add LIBRARY_PATH and PATH to `/etc/profile`: • export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/lib/sap" • export PATH=$PATH:/opt/universal/opscli/bin/:/opt/universal/usap/bin/:/root/startup/

Preparation of the RFC libraries for SUSE Linux

Step 1	Download the appropriate tar file for the target platform: `SAP_NW_RFC_Lin_x64.tar` or `SAP_NW_RFC_Lin_x86.tar` (64-bit and 32-bit respectively)
Step 2	Create the `/usr/local/lib/sap` directory for the SAP libraries to reside. ***REQUIRED, MUST MATCH THIS NAME:*** `md /usr/local/lib/sap`
Step 3	Copy the RFC tar file into the directory from Step 2 and extract the contents: `$tar -xvf SAP_NW_RFC_Lin_x64.tar`
Step 4	Add the path from Step 2 to the LD_LIBRARY_PATH environment variable `export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/lib/sap"`
Step 5	To maintain the LD_LIBRARY_PATH setting across sessions, place the export line from Step 4 into your profile. Example: Add LD_LIBRARY_PATH to `/etc/profile.local`. Note If `/etc/profile.local` does not exist, create it and exit from the user session to re-source the new profile.

File Content of profile.local

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/lib/sap"
PATH=$PATH:/opt/universal/ops/bin
PATH=$PATH:/opt/universal/ucmdmgr/bin
PATH=$PATH:/opt/universal/usap/bin
PATH=$PATH:/opt/universal/bin
export PATH
```

z/OS Installation

- Introduction
- Installation Checklist
- Detailed Information

Introduction

Note

These pages describe the installation of Universal Agent on a z/OS operating system. Unless otherwise specified, all references to Universal Agent for z/OS refer to version 7.2.x.

Updates to the content and format of the distribution file were delivered in the 6.5.0.0 Universal Agent release. The Installation Checklist below was updated to reflect those changes.

Universal Agent for z/OS is provided as an easily installed SMP/E package. After the SMP/E installation steps are complete, z/OS must be configured to meet the product requirements. Lastly, the individual product components must be customized to meet local production requirements.

The z/OS package includes the components for both Universal Agent for z/OS and optional components for Universal Agent for z/OS UNIX System Services (USS). For information on installation of Universal Agent for z/OS USS, see z/OS USS Installation.

Note

Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Agent component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

Installation Checklist

The following installation checklist provides an overview of the z/OS installation steps. The steps refer to pages that describe the procedures in detail. All z/OS installation pages should be read to avoid product installation and configuration problems.

The installation itself consists of running a number of batch jobs. The output of these batch jobs should be kept until a correct installation has been verified.

Step 1	Download one of the Universal Agent 7.2.x for z/OS product distribution files.
Step 2a	If one of the distribution files with the .tar.Z extension was downloaded, decompress it using any utility that can process files in Lempel-Ziv (that is, **compress**) format. Use **tar** to extract the files from the uncompressed file. To do both in a single statement, decompress the file to stdout and pipe the result to **tar** (for example, **zcat sb-v.r.m.l-zos.tar.Z | tar xvf -)**.
Step 2b	If one of the distribution files with the .zip extension was downloaded, decompress it using any utility that can process files in **ZIP** (or, PKZIP) format. The .zip file can also be extracted using Windows native file extraction tools.
Step 3	Modify the UNVFTP.TXT script provided by the distribution file. This script contains statements to upload the installation files and sample JCL, which can be used to process the files after upload. Two versions of these files are included in the distribution file. The files in the main directory are EBCDIC-encoded files. ASCII-encoded versions of these files are in the `./ascii` subdirectory. See z/OS Installation - Transferring Installation Files to z/OS for complete details.
Step 4	Modify the JCL uploaded from UNVRECV.JCL and run the job to receive the TSO XMIT file(s). If the distribution file that contains the DFDSS archive is used, UNVRECV.JCL also contains a step to restore the contents of the archive. See Receive the Installation Datasets for complete details.
Step 5	Edit the INSTALL(#SETUP) member to meet local requirements as described in the #SETUP comment prolog. #SETUP creates customized installation JCL members in the INSTALL library. Execute the installation jobs as required for your SMP/E environment. See z/OS Installation - SMPE Installation for complete details.
Step 6	Create started task user and group profiles for Universal Broker and Universal Enterprise Controller. Seez/OS Configuration - Started Tasks for complete details.
Step 7	Define the SUNVLOAD library to APF, Program Control, and LNKLST. See z/OS Configuration - Load Library for complete details.
Step 8	Define SMF exits for Universal Command and Universal Automation Center Agent. See z/OS Configuration - SMF Exits for complete details.
Step 9	Configure the UNIX System Services file systems used by the Universal Broker and Universal Enterprise Controller. See Universal Agent Database Configuration for complete details.
Step 10	Perform individual component customization, including JCL procedure edits and adding component licenses. See z/OS Installation - Customization for complete details.
Step 11	Follow the optional steps described in z/OS USS Installation to complete the installation, if desired.

Detailed Information

The following pages provide detailed information for z/OS Installation:

- Installation Package
- Installation Requirements
- Distribution File
- Transferring Installation Files to zOS
- SMPE Installation
- Configuration
- Customization
- Cumulative PTF Maintenance
- Time Zone Environment Variable
- TCPIP Configuration
- SAP RFC DLL
- Configuration of zOS System SSL
- Configuration of Security
- Performance Guidelines
- Converting STC User Profiles to a Non-Zero UID
- Data Set Inventory
- Licensing

z/OS Installation - Installation Package

- Package Components
- Component Compatibility

Package Components

The Universal Agent for z/OS package contains the following Universal Agent for z/OS components:

- Universal Broker 7.2.x
- Universal Automation Center Agent 7.2.x
- Universal Certificate 7.2.x
- Universal Command Manager and Server 7.2.x
- Universal Connector for PeopleSoft 7.2.x
- Universal Connector for SAP 6.2.x
- Universal Control Manager and Server 7.2.x
- Universal Controller Command Line Interface (CLI) 7.2.x
- Universal Data Mover Manager and Server 7.2.x
- Universal Encrypt 7.2.x
- Universal Enterprise Controller 7.2.x
- Universal Event Monitor Manager 7.2.x
- Universal FTP Client 7.2.x
- Universal Message to Exit Code Translator 7.2.x
- Universal Query 7.2.x

Note

For the list of Universal Agent for z/OS USS components included in the z/OS package, see z/OS USS Installation - Installation Package.

Component Compatibility

The following table identifies the compatibility of Universal Agent for z/OS components with previous component / product versions.

Component	Compatibility
Universal Broker 7.2.x	Workload Automation / Stonebranch Solutions / Universal Products releases 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Command 7.2.x	Universal Command 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 7.2.x	Universal Control 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 7.2.x	Universal Data Mover 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Encrypt 7.2.x	Universal Encrypt 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 7.2.x	Universal Broker 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Enterprise Controller 7.2.x	Not compatible with previous versions of Universal Enterprise Controller Client Applications.
Universal Event Monitor 7.2.x	Universal Event Monitor 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

z/OS Installation - Installation Requirements

- System Requirements
- Data Set Space Requirements
- Platform Requirements

System Requirements

Universal Agent for z/OS requires the following software releases:

- z/OS 2.2, 2.3, or 2.4.
- SMP/E 3.5 or later.
- IBM Communication Server for z/OS 2.2, 2.3, or 2.4.
- IBM Language Environment (LE) for z/OS 2.2, 2.3, or 2.4.
- Workstation capable of establishing a TCP/IP network connection to the z/OS system.
- TSO user ID with an OMVS segment.
- About 1900 cylinders of DASD.
- Two available TCP/IP ports on z/OS.

All Universal Agent programs use z/OS UNIX System Services. As such, z/OS UNIX requires the user profile with which a program executes to have a properly defined OMVS segment. The OMVS segment should define a unique UID value. The HOME value must specify an existing home directory to which the user ID has read and write access.

Additionally, the group(s) that the user ID is associated with must have an OMVS segment that defines a unique GID value for the group. Refer to IBM's UNIX System Services Planning manual for additional details on defining z/OS UNIX users.

Data Set Space Requirements

As part of the Universal Agent for z/OS package installation, a number of SMP/E and non-SMP/E data sets are allocated and cataloged. The space requirements for these data sets are listed in z/OS Installation - Data Set Inventory.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

z/OS Installation - Distribution File

- z/OS Distribution File
- Obtaining the Distribution File
- Distribution File Format
  - Compressed Archive
  - ZIP Archive
- Distribution File Contents
  - DFDSS Backup Distribution File
  - Expanded Format Distribution File

z/OS Distribution File

The Universal Agent for z/OS product distribution file contains all of the files required for the installation of the Universal Agent for z/OS package.

Obtaining the Distribution File

To obtain the Universal Agent for z/OS package, download the corresponding product distribution file from the Stonebranch Customer Portal.

A customer user name and password — provided by Stonebranch, Inc. — are required to access this area.

Note

If you are installing a Cumulative PTF Maintenance to a Universal Agent for z/OS installation, you must download the z/OS PTFs package from the Stonebranch Customer Portal.

Distribution File Format

The z/OS Installation's distribution file is available in two formats:

1. Compressed archive
2. Compressed .zip file

The contents of both files are the same. The different formats are provided for flexibility.

Compressed Archive

Two different compressed archive distribution files are provided.

One version contains a DFDSS backup of the SMP/E install datasets. The distribution file is a compressed TAR file that contains that backup in TSO TRANSMIT format along with supporting files used to upload and receive the backup.

This file uses the following naming convention:

`sb-Version.Release.Modification Level.Maintenance Level-operating system.tar.Z`

For example: **`sb-7.2.0.0-zos.tar.Z`**

The other version is intended for users who do not use DFDSS. It also is provided for users with RACF restrictions that prohibit restoring the SBI.** datasets contained in the DFDSS backup. The expanded contents in the TAR file provided by this distribution file contains each of the SMP/E install datasets in TSO TRANSMIT format. Each dataset can be allocated and uploaded using the sample UNVFTP.TXT. The JCL provided in UNVRECV.JCL demonstrates how to perform a TSO RECEIVE on each of the uploaded datasets.

This file uses the following naming convention:

`sb-Version.Release.Modification Level.Maintenance Level-zos-EXP.tar.Z`

For example: **`sb-7.2.0.0-zos-EXP.tar.Z`**

ZIP Archive

The ZIP versions of the z/OS Universal Agent distribution files have the same contents as their compressed archive counterparts and use the following naming convention:

- **`sb-Version.Release.Modification Level.Maintenance Level-operating system.zip`**

- **sb-Version.Release.Modification Level.Maintenance Level-operating system-EXP.zip**

For example:

- **sb-7.2.0.0-zos.zip**
- **sb-7.2.0.0-zos-EXP.zip**

Distribution File Contents

DFDSS Backup Distribution File

The following table lists the files included in the Universal Agent for z/OS distribution file that delivers the SMP/E install datasets in a DFDSS backup (see Distribution File Format).

File Name	Description
README.TXT	Contains instructions for preparing and executing the install. Note This is an EBCDIC-encoded text file.
UNVFTP.TXT	Sample FTP script that allocates and transfers datasets used to store the distribution file's contents on the z/OS system. Note This is an EBCDIC-encoded text file.
UNVRECV.JCL	Sample JCL that may be used to execute a TSO RECEIVE on the uploaded UNV720.XMT file, perform a DFDSS RESTORE on the received dataset, and unload the contents of dataset that holds the installation jobs. Note This is an EBCDIC-encoded text file.
UNV720.XMT	File in TSO TRANSMIT format that contains a DFDSS backup of the datasets required to install Universal Agent for z/OS.
ascii/README.TXT	ASCII-encoded copy of README.TXT.
ascii/UNVFTP.TXT	ASCII-encoded copy of UNVFTP.TXT.
ascii/UNVRECV.JCL	ASCII-encoded copy of UNVRECV.JCL.

Expanded Format Distribution File

The following table lists the files included in the Universal Agent for z/OS distribution file that delivers each SMP/E install dataset in TSO TRANSMIT format (see Distribution File Format).

File Name	Description
README.TXT	Contains instructions for preparing and executing the install. Note This is an EBCDIC-encoded text file.
UNVFTP.TXT	Sample FTP script that allocates and transfers UNVRECV.JCL and the SMP/E install datasets to the z/OS system. Note This is an EBCDIC-encoded text file.

UNVRECV.JCL	Sample JCL that may be used to execute a TSO RECEIVE on the uploaded *.XMT files. Note This is an EBCDIC-encoded text file.
INSTALL.XMT	Universal Agent package installation JCL.
SMPMCS.XMT	Universal Agent SMP/E MCS statements.
UAG720F1.XMT	Universal Automation Center Agent SMP/E FMID TUAG720 relative file 1.
UAG720F2.XMT	Universal Automation Center Agent SMP/E FMID TUAG720 relative file 2.
UAG720F3.XMT	Universal Automation Center Agent SMP/E FMID TUAG720 relative file 3.
UBR720F1.XMT	Universal Broker SMP/E FMID TUBR720 relative file 1.
UBR720F2.XMT	Universal Broker SMP/E FMID TUBR720 relative file 2.
UBR720F3.XMT	Universal Broker SMP/E FMID TUBR720 relative file 3.
UCM720F1.XMT	Universal Command SMP/E FMID TUCM720 relative file 1.
UCM720F2.XMT	Universal Command SMP/E FMID TUCM720 relative file 2.
UCM720F3.XMT	Universal Command SMP/E FMID TUCM720 relative file 3.
UDM720F1.XMT	Universal Data Mover SMP/E FMID TUDM720 relative file 1.
UDM720F2.XMT	Universal Data Mover SMP/E FMID TUDM720 relative file 2.
UDM720F3.XMT	Universal Data Mover SMP/E FMID TUDM720 relative file 3.
UEC720F1.XMT	Universal Enterprise Controller SMP/E FMID TUEC720 relative file 1.
UEC720F2.XMT	Universal Enterprise Controller SMP/E FMID TUEC720 relative file 2.
UEM720F1.XMT	Universal Event Monitor SMP/E FMID TUEM720 relative file 1.
UEM720F2.XMT	Universal Event Monitor SMP/E FMID TUEM720 relative file 2.
UEM720F3.XMT	Universal Event Monitor SMP/E FMID TUEM720 relative file 3.
UNV720F1.XMT	Universal Common SMP/E FMID TUNV720 relative file 1.
USD620F1.XMT	SAP RFC DLL FMID TUSD620 relative file 1.
USP620F1.XMT	Universal Connector for SAP SMP/E FMID TUSP620 relative file 1.
USP620F1.XMT	Universal Connector for SAP SMP/E FMID TUSP620 relative file 2.
USP620F1.XMT	Universal Connector for SAP SMP/E FMID TUSP620 relative file 3.
UTL720F1.XMT	Universal Utilities SMP/E FMID TUTL720 relative file 1.
UTL720F1.XMT	Universal Utilities SMP/E FMID TUTL720 relative file 2.
UXD720F1.XMT	IBM XML Toolkit FMID TUXD720 relative file 1.
UBR720U1.XMT	USS Universal Broker FMID UUBR720 relative file 1.
UBR720U2.XMT	USS Universal Broker FMID UUBR720 relative file 2.
UCM720U1.XMT	USS Universal Broker FMID UUCM720 relative file 1.
UCM720U2.XMT	USS Universal Broker FMID UUCM720 relative file 2.
UDM720U1.XMT	USS Universal Broker FMID UUDM720 relative file 1.
UDM720U2.XMT	USS Universal Broker FMID UUDM720 relative file 2.
UEM720U1.XMT	USS Universal Broker FMID UUEM720 relative file 1.
UEM720U2.XMT	USS Universal Broker FMID UUEM720 relative file 2.
UNV720U1.XMT	USS Universal Broker FMID UUBR720 relative file 1.
UTL720U1.XMT	USS Universal Broker FMID UUBR720 relative file 1.

ascii/README.TXT	ASCII-encoded copy of README.TXT.
ascii/UNVFTP.TXT	ASCII-encoded copy of UNVFTP.TXT.
ascii/UNVRECV.JCL	ASCII-encoded copy of UNVRECV.JCL.

z/OS Installation - Transferring Installation Files to z/OS

- Overview
- Transferring the Installation Files
- Extract the Installation Files
  - Compressed Archive
  - ZIP Archive
- Transfer the Installation Files
  - DFDSS Backup Distribution File
  - Expanded Format Distribution File
- Receive the Installation Datasets
  - DFDSS Backup Distribution File
  - Expanded Format Distribution File

Overview

After downloading the distribution file to your workstation, you must extract the files and transfer them to the z/OS system, upon which Universal Agent for z/OS will be installed.

Transferring the Installation Files

Follow the steps below to transfer the contents of the distribution file to the z/OS system.

1	Extract the installation files from the distribution file.
2	Transfer the Installation Files.
3	Receive the installation dataset(s).

Extract the Installation Files

Each version of the Universal Agent for z/OS Installation distribution file is available in two formats:

1. Compressed archive
2. Compressed .zip file

The contents of both files are the same. The different formats are provided for flexibility.

Compressed Archive

This file may be decompressed using any utility that can process files in Lempel-Ziv (that is, **compress**) format. The contents of the decompressed file can be extracted using **tar**.

For example, use the following to decompress and extract the 7.2.0.0 Universal Agent for z/OS distribution file that contains the DFDSS backup on a UNIX-based system:

```
zcat sb-7.2.0.0-zos.tar.Z | tar xvf -
```

Similarly, use the following to decompress and extract the 7.2.0.0 Universal Agent for z/OS distribution file that contains the expanded archive of SMP/E installation datasets:

```
zcat sb-7.2.0.0-EXP-zos.tar.Z | tar xvf -
```

ZIP Archive

This file (for example, **`sb-7.2.0.0-zos.zip`**) can be decompressed using any utility that can process files in **ZIP** (or PKZIP) format. To unzip the file, use a third-party ZIP-based extraction application or the native Windows file extraction tool available from Windows Explorer. For example, to extract the files using Windows' File Explorer (or just Windows Explorer in some older Windows versions), right-click on the file name, select Extract All from the pop-up menu, and follow the instructions.

Transfer the Installation Files

For the .XMT file(s), a binary transfer must be performed; otherwise, the XMIT dataset(s) will not unpack correctly. For the .JCL file, text transfer must be performed.

Due to the numerous file transfer products available on the market, it is impractical to provide instructions for them all. FTP is one of the most popular, and the UNVFTP.TXT file included in the distribution file is a sample script that can be used as a reference to customize the transfer to your environment.

DFDSS Backup Distribution File

If you want to use UNVFTP.TXT (see the following illustration), begin by making the following changes:

Step 1	Change #HOSTNAME to the z/OS host name or IP address.
Step 2	Change #USERNAME to a z/OS user ID that has updated access to the XMIT datasets being updated.
Step 3	Change #PASSWORD to the z/OS user ID's password.
Step 4	Change #HLQ to the data set high level qualifier used to allocate the XMIT datasets.

```
open #HOSTNAME
user #USERNAME #PASSWORD

ascii
quote site blksize=3120 recfm=fb lrecl=80 pri=1 sec=1 cyl
put UNVRECV.JCL '#HLQ.UNV.V7R2M0.RECV.JCL'

;Uncomment these lines when transferring from a
;Unix- or Windows-based system.
;put ascii/UNVRECV/JCL '#HLQ.UNV.V7R2M0.RECV.JCL'

bin
quote site blksize=3120 recfm=fb lrecl=80 pri=1000 sec=100 cyl
put UNV720.XMT '#HLQ.UNV.V7R2M0.DIST.XMIT'

quit
```

Execute the script with FTP using whatever options the FTP client supports for reading command files. Most FTP clients support redirection from STDIN.

Carefully review the FTP output to confirm that each of the files are transferred successfully.

After uploading, review the comments contained in `'#HLQ.UNV.V7R2M0.RECV.JCL'` and make the necessary changes. Submit the job to receive and restore the install file.

Expanded Format Distribution File

The steps to transfer the Expanded Format version of the distribution file are the same as for the DFDSS Backup distribution file (see the following illustration). However, the UNVFTP.TXT sample provided in that file allocates and uploads each SMP/E install file separately.

Step 1	Change #HOSTNAME to the z/OS host name or IP address.
Step 2	Change #USERNAME to a z/OS user ID that has updated access to the XMIT datasets being updated.
Step 3	Change #PASSWORD to the z/OS user ID's password.
Step 4	Change #HLQ to the data set high level qualifier used to allocate the XMIT datasets.
Step 5	Optional; Change #VOLSER to a volume upon which the *.XMT files provided in the Expanded Format version of the distribution file should reside.

```
open #HOSTNAME
user #USERID #PSWD

ascii
quote site blksize=3120 recfm=fb lrecl=80 pri=1 sec=1 track
put UNVRECV.JCL '#HLQ.UNV.V7R2M0.RECV.JCL'

;Uncomment these lines when transferring from a
```

```
;Unix- or Windows-based system.
;put ascii/UNVRECV.JCL '#HLQ.UNV.V7R2M0.RECV.JCL'

cd '#HLQ.UNV.V6R7M0'

bin
quote site blksize=3120 recfm=fb lrecl=80 cyl unit=sysallda vol=#VOLSER

quote site pri=1 sec=1
del INSTALL.XMIT
put INSTALL.XMT INSTALL.XMIT

quote site pri=10 sec=10
del SMPMCS.XMIT
put SMPMCS.XMT SMPMCS.XMIT

quote site pri=1 sec=1
del TUNV720.F1.XMIT
put UNV720F1.XMT TUNV720.F1.XMIT

quote site pri=1
del TUTL720.F1.XMIT
put UTL720F1.XMT TUTL720.F1.XMIT

quote site pri=70
del TUTL720.F2.XMIT
put UTL720F2.XMT TUTL720.F2.XMIT

quote site pri=1
del TUBR720.F1.XMIT
put UBR720F1.XMT TUBR720.F1.XMIT

quote site pri=1
del TUBR720.F2.XMIT
put UBR720F2.XMT TUBR720.F2.XMIT

quote site pri=120
del TUBR720.F3.XMIT
put UBR720F3.XMT TUBR720.F3.XMIT

quote site pri=1
del TUCM720.F1.XMIT
put UCM720F1.XMT TUCM720.F1.XMIT

quote site pri=1
del TUCM720.F2.XMIT
put UCM720F2.XMT TUCM720.F2.XMIT

quote site pri=45
del TUCM720.F3.XMIT
put UCM720F3.XMT TUCM720.F3.XMIT

quote site pri=1
del TUDM720.F1.XMIT
put UDM720F1.XMT TUDM720.F1.XMIT

quote site pri=1
del TUDM720.F2.XMIT
put UDM720F2.XMT TUDM720.F2.XMIT

quote site pri=55
del TUDM720.F3.XMIT
put UDM720F3.XMT TUDM720.F3.XMIT

quote site pri=1
del TUEM720.F1.XMIT
put UEM720F1.XMT TUEM720.F1.XMIT

quote site pri=1
del TUEM720.F2.XMIT
put UEM720F2.XMT TUEM720.F2.XMIT

quote site pri=55
del TUEM720.F3.XMIT
put UEM720F3.XMT TUEM720.F3.XMIT

quote site pri=120
del TUXD720.F1.XMIT
put UXD720F1.XMT TUXD720.F1.XMIT

quote site pri=1
del TUAG720.F1.XMIT
put UAG720F1.XMT TUAG720.F1.XMIT
```

```
quote site pri=1
del TUAG720.F2.XMIT
put UAG720F2.XMT TUAG720.F2.XMIT

quote site pri=40
del TUAG720.F3.XMIT
put UAG720F3.XMT TUAG720.F3.XMIT

quote site pri=1
del TUEC720.F1.XMIT
put UEC7200F1.XMT TUEC720.F1.XMIT

quote site pri=70
del TUEC720.F2.XMIT
put UEC720F2.XMT TUEC720.F2.XMIT

quote site pri=20
del TUSD620.F1.XMIT
put USD620F1.XMT TUSD620.F1.XMIT

quote site pri=1
del TUSP620.F1.XMIT
put USP620F1.XMT TUSP620.F1.XMIT

quote site pri=1
del TUSP620.F2.XMIT
put USP620F2.XMT TUSP620.F2.XMIT

quote site pri=15
del TUSP620.F3.XMIT
put USP620F3.XMT TUSP620.F3.XMIT

quote site pri=1
del UUNV720.F1.XMIT
put UNV720U1.XMT UUNV720.F1.XMIT

quote site pri=1
del UUBR720.F1.XMIT
put UBR720U1.XMT UUBR720.F1.XMIT

quote site pri=50
del UUBR720.F2.XMIT
put UBR720U2.XMT UUBR720.F2.XMIT

quote site pri=1
del UUCM720.F1.XMIT
put UCM720U1.XMT UUCM720.F1.XMIT

quote site pri=30
del UUCM720.F2.XMIT
put UCM720U2.XMT UUCM720.F2.XMIT

quote site pri=1
del UUDM720.F1.XMIT
put UDM720U1.XMT UUDM720.F1.XMIT

quote site pri=30
del UUDM720.F2.XMIT
put UDM720U2.XMT UUDM720.F2.XMIT

quote site pri=1
del UUEM720.F1.XMIT
put UEM720U1.XMT UUEM720.F1.XMIT

quote site pri=30
del UUEM720.F2.XMIT
put UEM720U2.XMT UUEM720.F2.XMIT

quote site pri=70
del UUTL720.F1.XMIT
put UTL720U1.XMT UUTL720.F1.XMIT

quit
```

Execute the script with FTP using whatever options the FTP client supports for reading command files. Most FTP clients support redirection from STDIN.

Carefully review the FTP output to confirm that each of the files are transferred successfully.

After uploading, review the comments contained in `'#HLQ.UNV.V6R7M0.RECV.JCL'` and make the necessary changes. Submit the job to receive and restore the install file.

Receive the Installation Datasets

DFDSS Backup Distribution File

Note

The following instructions apply to the version of the distribution file that contains the DFDSS backup. See Expanded Format Distribution File, below, for information on processing the other version of the distribution file.

After transferring the installation file to z/OS, you must unpack them using the JCL uploaded from UNVRECV.JCL (which should now reside in `#HLQ.UNV.V6R7M0.RECV.JCL`). The following illustration shows the statements contained in UNVRECV.JCL. The job performs a TSO RECEIVE on the UNV720.XMT file uploaded by UNVFTP.TXT. It then executes a DFDSS restore on the transmitted data set to extract the packaged files.

The JCL includes a MODIFICATIONS section at the beginning of the file that describes what JCL modifications are required prior to submitting the job. Read and complete each of the listed modifications.

Submit the job on z/OS. Return code 0 is expected.

After the data set is successfully unpacked, it is longer required and can be deleted. The files then are ready to be installed.

```
//UNVRECV  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//            COND=(0,NE),REGION=(0M)
//*******************************************************************
//* (C) COPYRIGHT 2000-2020 STONEBRANCH, INC.  ALL RIGHTS RESERVED.
//*
//* STONEBRANCH, INC.
//* UNIVERSAL PRODUCTS
//*
//* UNVRECV
//*
//* DESCRIPTION
//* -----------
//* PERFORM A TSO RECEIVE ON THE TRANSMITTED DISTRIBUTION DATA SET
//* AND A DFHSMdss RESTORE TO PREPARE FOR INSTALLATION.
//*
//* ALL STEPS MUST END WITH RC 0.
//*
//* MODIFICATIONS
//* -------------
//* 1  MODIFY THE JOB STATEMENT TO MEET LOCAL REQUIREMENTS.
//*
//* 2  CHANGE ALL '#HLQ' TO THE HIGH-LEVEL QUALIFIER OF THE
//*    UNIVERSAL PRODUCTS DATA SETS.
//*
//* 3  CHANGE ALL '#VOLSER' TO THE VOLUME SERIAL NAME ON WHICH TO
//*    ALLOCATE THE DISTRIBUTION DATA SETS.
//*******************************************************************
//*
//DEL00    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE #HLQ.UNV.V7R2M0.DIST
  SET MAXCC=0
/*
//*
//*******************************************************************
//* RECEIVE DISTRIBUTION FILE
//*******************************************************************
//RECV     EXEC PGM=IKJEFT01
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTERM  DD  SYSOUT=*
//SYSTSIN  DD  *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.DIST.XMIT')
    DA('#HLQ.UNV. V7R2M0.DIST') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    CYL SPACE(750 100) RELEASE
/*
//*
//DEL01    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE #HLQ.UNV.V7R2M0.INSTALL.UNLD
  DELETE #HLQ.UNV.V7R2M0.SMPMCS
  DELETE #HLQ.UNV.V7R2M0.TUAG720.*
```

```
  DELETE #HLQ.UNV.V7R2M0.TUBR720.*
  DELETE #HLQ.UNV.V7R2M0.TUCM720.*
  DELETE #HLQ.UNV.V7R2M0.TUDM720.*
  DELETE #HLQ.UNV.V7R2M0.TUEC720.*
  DELETE #HLQ.UNV.V7R2M0.TUEM720.*
  DELETE #HLQ.UNV.V7R2M0.TUNV720.*
  DELETE #HLQ.UNV.V7R2M0.TUSD620.*
  DELETE #HLQ.UNV.V7R2M0.TUSP620.*
  DELETE #HLQ.UNV.V7R2M0.TUTL720.*
  DELETE #HLQ.UNV.V7R2M0.TUXD720.*
  DELETE #HLQ.UNV.V7R2M0.UUBR720.*
  DELETE #HLQ.UNV.V7R2M0.UUCM720.*
  DELETE #HLQ.UNV.V7R2M0.UUDM720.*
  DELETE #HLQ.UNV.V7R2M0.UUEM720.*
  DELETE #HLQ.UNV.V7R2M0.UUNV720.*
  DELETE #HLQ.UNV.V7R2M0.UUTL720.*
  SET MAXCC=0
/*
//*
//*******************************************************************
//* RESTORE BACKUP
//*******************************************************************
//RESTORE  EXEC PGM=ADRDSSU
//IN1      DD DISP=SHR,
//            DSN=#HLQ.UNV.V7R2M0.DIST
//OUT1     DD DISP=SHR,
//            UNIT=SYSALLDA,VOL=SER=#VOLSER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  RESTORE INDD(IN1) -
          DS(INCL(**)) -
          OUTDD(OUT1) -
          CATALOG -
          RENAMEUNCONDITIONAL(SBI.**,#HLQ.**)
/*
//*
//DEL02    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE #HLQ.UNV.V7R2M0.INSTALL
  SET MAXCC=0
/*
//*
//*******************************************************************
//* RELOAD INSTALL JCL LIBRARY FROM SEQUENTIAL BACKUP
//*******************************************************************
//RELOAD   EXEC PGM=IEBCOPY
//IN1      DD DISP=OLD,
//            DSN=#HLQ.UNV.V7R2M0.INSTALL.UNLD
//OUT1     DD DSN=#HLQ.UNV.V7R2M0.INSTALL,
//            DISP=(,CATLG),
//            DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3120),
//            SPACE=(TRK,(15,10,10)),
//            UNIT=SYSALLDA,VOL=SER=#VOLSER
//SYSUT3   DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT4   DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  COPY I=(IN1),O=OUT1
/*
//*
```

Expanded Format Distribution File

Note

The following instructions apply to the Expanded Format version of the distribution file that contains the each of the SMP/E datasets separately. See DFDSS Backup Distribution File, above, for information on processing the version of the distribution file that provides the SMP/E install datasets packaged inside a DFDSS backup.

After transferring the installation files to z/OS, you must receive them using the JCL uploaded from UNVRECV.JCL (which should now reside in `#HLQ.UNV.V6R7M0.RECV.JCL`). The following illustration shows the statements contained in UNVRECV.JCL. The job performs a TSO RECEIVE on each of the files uploaded by UNVFTP.TXT.

The JCL includes a MODIFICATIONS section at the beginning of the file that describes what JCL modifications are required prior to submitting the job. Read and complete each of the listed modifications.

Submit the job on z/OS. Return code 0 is expected.

After the datasets are successfully received, the *.XMIT files are no longer required and can be deleted. The files then are ready to be installed.

```
//UNVRECV  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//             COND=(0,NE),REGION=(0M)
//*******************************************************************
//* (C) COPYRIGHT 2000-2020 STONEBRANCH, INC.  ALL RIGHTS RESERVED.
//*
//* STONEBRANCH, INC.
//* UNIVERSAL PRODUCTS
//*
//* UNVRECV
//*
//* DESCRIPTION
//* -----------
//* PERFORM A TSO RECEIVE ON THE TRANSMITTED DISTRIBUTION DATA SETS.
//* ALL STEPS MUST END WITH RC 0.
//*
//* MODIFICATIONS
//* -------------
//* 1  MODIFY THE JOB STATEMENT TO MEET LOCAL REQUIREMENTS.
//*
//* 2  CHANGE ALL '#HLQ' TO THE HIGH-LEVEL QUALIFIER OF THE
//*    UNIVERSAL PRODUCTS DATA SETS.
//*
//* 3  CHANGE ALL '#VOLSER' TO THE VOLUME SERIAL NAME ON WHICH TO
//*    ALLOCATE THE DISTRIBUTION DATA SETS.
//*******************************************************************
//*
//DEL01    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE #HLQ.UNV.V7R2M0.INSTALL
  DELETE #HLQ.UNV.V7R2M0.SMPMCS
  DELETE #HLQ.UNV.V7R2M0.TUAG720.*
  DELETE #HLQ.UNV.V7R2M0.TUBR720.*
  DELETE #HLQ.UNV.V7R2M0.TUCM720.*
  DELETE #HLQ.UNV.V7R2M0.TUDM720.*
  DELETE #HLQ.UNV.V7R2M0.TUEC720.*
  DELETE #HLQ.UNV.V7R2M0.TUEM720.*
  DELETE #HLQ.UNV.V7R2M0.TUNV720.*
  DELETE #HLQ.UNV.V7R2M0.TUSD620.*
  DELETE #HLQ.UNV.V7R2M0.TUSP620.*
  DELETE #HLQ.UNV.V7R2M0.TUTL720.*
  DELETE #HLQ.UNV.V7R2M0.TUXD720.*
  DELETE #HLQ.UNV.V7R2M0.UUBR720.*
  DELETE #HLQ.UNV.V7R2M0.UUCM720.*
  DELETE #HLQ.UNV.V7R2M0.UUDM720.*
  DELETE #HLQ.UNV.V7R2M0.UUEM720.*
  DELETE #HLQ.UNV.V7R2M0.UUNV720.*
  DELETE #HLQ.UNV.V7R2M0.UUTL720.*
/*
//*
//TSO      PROC
//S1       EXEC PGM=IKJEFT01
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTERM  DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//         PEND
//*
//SMPMCS   EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.SMPMCS.XMIT')
    DA('#HLQ.UNV.V7R2M0.SMPMCS') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(100 50) RELEASE
//INSTALL  EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.INSTALL.XMIT')
    DA('#HLQ.UNV.V7R2M0.INSTALL') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 10)
//UNV720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUNV720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUNV720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UTL720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUTL720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUTL720.F1') +
```

```
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UTL720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUTL720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUTL720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1000 10) RELEASE
//UBR720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUBR720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUBR720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UBR720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUBR720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUBR720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UBR720F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUBR720.F3.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUBR720.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1720 15) RELEASE
//UCM720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUCM720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUCM720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UCM720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUCM720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUCM720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UCM720F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUCM720.F3.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUCM720.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(600 15) RELEASE
//UDM720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUDM720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUDM720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUDM720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUDM720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM720F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUDM720.F3.XMIT')
    DA('#HLQ.UNV.VV7R2M0.TUDM720.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1000 15) RELEASE
//UEM720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.VV7R2M0.TUEM720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUEM720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEM720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUEM720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUEM720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEM720F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUEM720.F3.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUEM720.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(200 15) RELEASE
//UXD720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUXD720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUXD720.F1') +
```

```
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1530 15) RELEASE
//UAG720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUAG720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUAG720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UAG720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUAG720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUAG720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UAG720F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUAG720.F3.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUAG720.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(600 15) RELEASE
//UEC720F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUEC720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUEC720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEC720F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUEC720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUEC720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1000 15) RELEASE
//USD620F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUSD620.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUSD620.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(300 15) RELEASE
//USP620F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUSP620.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUSP620.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//USP620F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUSP620.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUSP620.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//USP620F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.TUSP620.F3.XMIT')
    DA('#HLQ.UNV.V7R2M0.TUSP620.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(200 15) RELEASE
//UBR720U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUBR720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUBR720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UBR720U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUBR720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUBR720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UCM720U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUCM720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUCM720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UCM720U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUCM720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUCM720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UDM720U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUDM720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUDM720.F1') +
```

```
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM720U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUDM720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUDM720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UEM720U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUEM720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUEM720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEM720U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUEM720.F2.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUEM720.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UTL720U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUTL720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUTL720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1000 15) RELEASE
//UNV720U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V7R2M0.UUNV720.F1.XMIT')
    DA('#HLQ.UNV.V7R2M0.UUNV720.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
```

z/OS Installation - SMPE Installation

- SMP/E Installation of Universal Agent for z/OS
- SMP/E
  - SMP/E FMIDs

SMP/E Installation of Universal Agent for z/OS

Universal Agent for z/OS is installed using SMP/E. The pages in this section describe how to perform SMP/E installation of the Universal Agent for z/OS package in a step-by-step process.

Six different installation processes are provided. The installation process that you use depends on the installation environment.

New Install, New CSI	Install any Universal Agent package from Stonebranch, Inc. for the first time, or install a Universal Agent package in a new SMP/E CSI.
Universal Agent 7.1.x Upgrade, Existing CSI	Upgrade a Universal Agent 7.1.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 7.1.x package.
Universal Agent 7.0.x Upgrade, Existing CSI	Upgrade a Universal Agent 7.0.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 7.0.x package.
Universal Agent 6.9.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.9.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.9.x package.
Universal Agent 6.8.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.8.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.8.x package.
Universal Agent 6.7.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.7.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.7.x package.
Universal Agent 6.6.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.6.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.6.x package.
Universal Agent 6.5.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.5.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.5.x package.
Universal Agent 6.4.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.4.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.4.x package.
Universal Agent 6.3.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.3.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.3.x package.
Universal Agent 6.2.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.2.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 6.2.x package.
Universal Agent 5.2.0 Upgrade, Existing CSI	Upgrade a Universal Agent 5.2.0 package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Agent 5.2.0 package.
Workload Automation 5.1.0 Upgrade, Existing CSI	Upgrade a Workload Automation 5.1.0 package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Workload Automation 5.1.0 package.
Stonebranch Solutions 4.x Upgrade, Existing CSI	Upgrade a Stonebranch Solutions 4.x package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Stonebranch Solutions 4.x package.
Universal Products 3.2.0 Upgrade, Existing CSI	Upgrade a Universal Products 3.2.0 package install. In this case, the Universal Agent package is installed into an SMP/E CSI that contains a Universal Products 3.2.0 package.
Universal Products 3.1.1 (and earlier) Upgrade, Existing CSI	To upgrade from Universal Products 3.1.1 or earlier, you first must upgrade to Universal Products 3.2.0, and then upgrade to Universal Agent 5.2.0. For information on upgrading to Universal Products 3.2.0, see the Universal Products Installation Guide in the Universal Products 3.2.0 Documentation set.

SMP/E

The Universal Agent for z/OS components are installed in the Universal Agent SMP/E CSI. This CSI should not be shared with any other vendor products; it should be used exclusively for Universal Agent.

Before making any changes to the SMP/E environment, back up the environment per your local procedures. IBM recommends backing up the entire SMP /E pack before any SMP/E installation begins. Two sample JCL members are provide in members **UNVBKUP** and **UNVREST**.

As of Universal Products version 2.2.0, all Universal Agent components share common SMP/E target and distribution libraries:

- Product ID is **UNV**.
- Last qualifier of target libraries start with **SUNV**.
- Last qualifier of distribution libraries start with **AUNV**.

SMP/E FMIDs

The following table identifies the SMP/E FMIDs for the Universal Agent for z/OS components. (For the list of SMP/E FMIDs for the Universal Agent for z /OS USS components, see z/OS USS Installation - Installation Requirements.)

Product	FMID	SMP/E Requisites
Universal Common 7.2.0	TUNV720	Supersedes and deletes FMID TUNV220, TUNV310, TUNV311, TUNV320, TUNV410, TUNV420, TUNV430, TUNV510, TUNV520, TUNV620, TUNV630, TUNV641, TUNV650, TUNV660, TUNV670, TUNV680, TUNV690, TUNV700, and TUNV710.
Universal Automation Center Agent 7.2.0	TUAG720	TUNV720 is a prerequisite.
Universal Broker 7.2.0	TUBR720	TUNV720 is a prerequisite. Supersedes and deletes FMID TUBR310, TUBR311, TUBR320, TUBR410, TUBR420, TUBR430, TUBR510, TUBR520, TUBR620, TUBR630, TUBR641, TUBR650, TUBR660, TUBR670, TUBR680, TUBR690, TUBR700, and TUBR710.
Universal Command 7.2.0	TUCM720	TUBR720 is a prerequisite. Supersedes and deletes FMID TUCM110, TUCM120, TUCM210, TUCM220, TUCM310, TUCM311, TUCM320, TUCM410, TUCM420, TUCM430, TUCM510, TUCM520, TUCM620, TUCM630, TUCM641, TUCM650, TUCM660, TUCM670, TUCM680, TUCM690, TUCM700, and TUCM710.
Universal Enterprise Controller 7.2.0	TUEC720	TUNV720 is a prerequisite. Supersedes and deletes FMID TUEC110, TUEC310, TUEC320, TUEC410, TUEC420, TUEC430, TUEC510, TUEC520, TUEC620, TUEC630, TUEC641, TUEC650, TUEC660, TUEC670, TUEC680, TUEC690, TUEC700, and TUEC710.
Universal Data Mover 7.2.0	TUDM720	TUBR720 is a prerequisite. Supersedes and deletes FMID TUDM110, TUDM310, TUDM311, TUDM320, TUDM410, TUDM420, TUDM430, TUDM510, TUDM520, TUDM620, TUDM630, TUDM641, TUDM650, TUDM660, TUDM670, TUDM680, TUDM690, TUDM700, and TUDM710.
SAP RFC DLL	TUSD720	TUNV720 is a prerequisite. Supersedes and deletes FMID TUSD310, TUSD311, TUSD320, TUSD410, TUSD420, TUSD430, TUSD510, TUSD520, TUSD521, TUSD522, TUSD541, TUSD650, TUSD660, TUSD670, TUSD680, TUSD690, TUSD700, and TUSD710.
Universal Connector	TUSP720	TUSD720 is a prerequisite. Supersedes and deletes FMID TUSP120, TUSP310, TUSP311, TUSP320, TUSP410, TYSP420, TUSP430, TUSP510, TUSP520, TUSP521, TUSP522, TUSP541, TUSP650, TUSP660, TUSP670, TUSP680, TUSP690, TUSP700, and TUSP710.
Universal Utilities	TUTL720	TUNV720 is a prerequisite. Supersedes and deletes FMID TUEN110, TUEN120, TUEN210, TUEN220, TUEN310, TUEN311, TUTL320, TUTL410, TUTL420, TUTL430, TUTL510, TUTL520, TUTL620, TUTL630, TUTL641 TUTL650, TUTL660, TUTL670, TUTL680, TUTL690, TUTL700, and TUTL710.
Universal Event Monitor	TUEM720	TUBR720 is a prerequisite. Supersedes and deletes FMID TUEM310, TUEM311, TUEM320, TUEM410, TUEM420, TUEM430, TUEM510, TUEM520, TUEM620, TUEM630, TUEM641, TUEM650, TUEM660, TUEM670, TUEM680, TUEM690, TUEM700, and TUEM710.
IBM XML Toolkit	TUXD720	TUNV720 is a prerequisite. Supersedes and deletes TUXD320, TUXD410, TUXD420, TUXD430, TUXD510, TUXD520, TUXD620, TUXD630, TUXD641, TUXD650, TUXD660, TUXD670, TUXD680, TUXD690, TUXD700, and TUXD710.

z/OS Installation - New Install, New CSI

New Install, New CSI

The New Install, New CSI installation process describes how to install the Universal Agent package in a newly allocated SMP/E CSI.

Use this installation process for either of these environments:

- Installing a Universal Agent package for the first time.
- Installing a Universal Agent package in a different SMP/E CSI than other Universal Agent components.

The installation JCL referenced by the following installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. If you want to change the customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. • All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN01**. The JCL allocates the SMP/E CSI data sets. • All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN02**. The JCL initializes the SMP/E CSI. • All steps must end with a return code 0.
Step 4	Submit the JCL in member **UNVIN03**. The JCL allocates Universal Agent target and distribution data sets and adds SMP/E **DDDEF** definitions to the Universal Agent CSI zones. • All steps must end with a return code 0.
Step 5	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. • All steps must end with a return code 0.

Step 6	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs.
	
	• Step APYFMID must end with a condition code of 0.
	• Step APYPTFS is considered successful under any of the following conditions:
	  ◦ Step ends with condition code 0.
	  ◦ Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**.
	  ◦ Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.

Step 7	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of the product FMIDs and any applied PTFs.
	
	• Step ACCFMID must end with a condition code of 0.
	• Step ACCPTFS is considered successful under any of the following conditions:
	  ◦ Step ends with condition code 0 or 4.
	  ◦ Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.

Step 8	Submit the JCL in member **UNVIN07**. The JCL member allocates product non-SMP/E data sets and databases, and formats the zFS data sets. • All steps must end with a return code 0.
Step 9	The product databases are allocated as z/OS UNIX zFS file systems. Additional product configuration is required to utilize zFS data sets. See Universal Agent Database Configuration for details. If you prefer to use HFS file systems, HFS data sets must be allocated. If you prefer to use the zFS data sets, skip this step. Submit the JCL in member **UNVINHF**. The JCL renames the zFS data sets and defines their HFS counterparts. • All steps must end with return code 0.

Step 10	Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries. • All steps must end with a return code 0.
Step 11	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. • All steps must end with a return code 0.
Step 12	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 7.1.x Upgrade, Existing CSI

Universal Agent 7.1.x Upgrade, Existing CSI

The Universal Agent 7.1.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 7.1.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 7.0.x Upgrade, Existing CSI

Universal Agent 7.0.x Upgrade, Existing CSI

The Universal Agent 7.0.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 7.0.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.9.x Upgrade, Existing CSI

Universal Agent 6.9.x Upgrade, Existing CSI

The Universal Agent 6.9.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.9.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.8.x Upgrade, Existing CSI

Universal Agent 6.8.x Upgrade, Existing CSI

The Universal Agent 6.8.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.8.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.7.x Upgrade, Existing CSI

Universal Agent 6.7.x Upgrade, Existing CSI

The Universal Agent 6.7.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.7.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.6.x Upgrade, Existing CSI

Universal Agent 6.6.x Upgrade, Existing CSI

The Universal Agent 6.6.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.6.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.5.x Upgrade, Existing CSI

Universal Agent 6.5.x Upgrade, Existing CSI

The Universal Agent 6.5.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.5.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.4.x Upgrade, Existing CSI

Universal Agent 6.4.x Upgrade, Existing CSI

The Universal Agent 6.4.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.4.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.3.x Upgrade, Existing CSI

Universal Agent 6.3.x Upgrade, Existing CSI

The Universal Agent 6.3.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.3.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Submit the JCL in member **UNVUG630**. The JCL allocates a PDS used by the Universal Broker keystore feature, installs the UFTP client configuration member UFTCFG00, and sets the location of the UFTP client in the UAG Server configuration member, UAGCFG00.
Step 7	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 6.2.x Upgrade, Existing CSI

Universal Agent 6.2.x Upgrade, Existing CSI

The Universal Agent 6.2.x Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.2.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Submit the JCL in member **UNVIN10**. The JCL creates the JSC VSAM dataset. All steps must end with a return code 0.
Step 7	Submit the JCL in member **UNVUG620**. The JCL allocates a PDS used by the Universal Broker keystore feature, installs the UFTP client configuration member UFTCFG00, and sets the location of the UFTP client and JSC dataset in the UAG Server configuration member, UAGCFG00.
Step 8	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Agent 5.2.0 Upgrade, Existing CSI

Universal Agent 5.2.0 Upgrade, Existing CSI

The Universal Agent 5.2.0 Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 5.2.0 package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Submit the JCL in member **UNVIN10**. The JCL creates the JSC VSAM dataset. All steps must end with a return code 0.
Step 7	Submit the JCL in member **UNVUG5XX**. The JCL allocates a PDS used by the Universal Broker keystore feature, installs the UFTP client configuration member UFTCFG00, and sets the location of the UFTP client and JSC dataset in the UAG Server configuration member, UAGCFG00.
Step 8	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Workload Automation 5.1.0 Upgrade, Existing CSI

Workload Automation 5.1.0 Upgrade, Existing CSI

The Workload Automation 5.1.0 Upgrade, Existing CSI installation process describes how to upgrade an existing Workload Automation 5.1.0 package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 6	Submit the JCL in member **UNVIN10**. The JCL creates the JSC VSAM dataset. All steps must end with a return code 0.
Step 7	Submit the JCL in member **UNVUG5XX**. The JCL allocates a PDS used by the Universal Broker keystore feature, installs the UFTP client configuration member UFTCFG00, and sets the location of the UFTP client and JSC dataset in the UAG Server configuration member, UAGCFG00.
Step 8	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Stonebranch Solutions 4.x Upgrade, Existing CSI

Stonebranch Solutions 4.x Upgrade, Existing CSI

The Stonebranch Solutions 4.x Upgrade, Existing CSI installation process describes how to upgrade an existing Stonebranch Solutions 4.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUG4XX**. The JCL removes the UAR configuration and component definition members in the non-SMP/E UNVCONF and UNVCOMP libraries, respectively, allocates the UAG model data set, and allocates a PDS used by the Universal Broker keystore feature. All steps must end with return code 0.
Step 3	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 4	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 5	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 6	Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries. All steps must end with a return code 0.
Step 7	Edit and submit the JCL in member**UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 8	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Universal Products 3.2.0 Upgrade, Existing CSI

Universal Products 3.2.0 Upgrade, Existing CSI

The Universal Products 3.2.0 Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Products 3.2.0 package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

Step 1	Edit and submit the JCL in member **#SETUP**. The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library. To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUG320**. The JCL upgrades the SMP/E Universal Products configuration from 3.2.0 to 5.2.0, and allocates the UAG model data set. The following changes are made: • Allocates a new AUNVHBIN SMP/E distribution data set, if necessary.
Step 3	Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 4	Submit the JCL in member **UNVIN05**. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 5	Submit the JCL in member **UNVIN06**. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 6	Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries. All steps must end with a return code 0.
Step 7	Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL. The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library. All steps must end with a return code 0.
Step 8	Perform required z/OS configuration steps as described in z/OS Installation - Configuration.

z/OS Installation - Configuration

Overview

The following pages describe the z/OS configuration steps required for the Universal Broker, Universal Enterprise Controller, and Universal Automation Center Agent:

- Started Tasks
- Load Library
- SMF Exits
- JES SYSOUT Processing
- Universal Agent Database Configuration

The Universal Broker component is a required component. Universal Enterprise Controller is optional. Universal Automation Center Agent is required to manage workload with Universal Controller.

The following configuration steps are described:

- Installing the started tasks and setting up the started task user IDs and group IDs.
- APF authorizing the SUNVLOAD library, adding it to program control and adding specific programs to the LNKLST.
- Installation of SMF exits.
- Configuration of Universal Spool and Universal Enterprise Controller HFS or zFS databases.

Member **UNVINRF** in the **INSTALL** library provides a sample JCL that can be used to execute the RACF commands described in the following pages as a TSO batch job.

Carefully read the comments in the member before submitting the JCL. The JCL is provided only as a sample. Follow local procedures as necessary for defining and altering security resources.

z/OS Configuration - Started Tasks

- z/OS Configuration - Started Tasks
- Started Task Security
- Universal Broker
  - Universal Broker User and Group Profiles
  - Universal Broker Data Access
- Universal Enterprise Controller
  - Universal Enterprise Controller User and Group Profiles
  - Universal Enterprise Controller Data Access
- Associate Started Tasks with User and Group Profiles
  - STARTED Class Profile
  - Started Procedures Table

z/OS Configuration - Started Tasks

The Universal Agent for z/OS solution consists of two started tasks:

1. Universal Broker is a required started task that provides a number of services for manager and server components.
2. Universal Enterprise Controller is an optional started task that provides monitoring and administration services.

The following started task JCL procedures are provided in the **SUNVSAMP** library:

- **UBROKER** is the JCL procedure for the Universal Broker started task.
- **UECTLR** is the JCL procedure for the Universal Enterprise Controller started task.

The JCL procedures are copied to a system procedure library by installation job **UNVIN09**. If this was not done, copy the JCL members to the appropriate procedure library for your local environment.

The started task programs utilize both z/OS UNIX System Services (USS) and MVS services. As a result of using USS services, the programs execute as USS processes. As do all USS processes, the Universal Broker and Universal Enterprise Controller processes must execute with user profiles that have a properly defined OMVS segments. Additionally, the user profiles must be permitted security access to privileged USS services in order for them to perform specific functions.

This page describes the following configuration tasks:

- Started task security requirements.
- Defining the started task user and group profiles.
- Permitting the started task user profile to required security resources.
- Associating started tasks with the user and group profiles.

Started Task Security

z/OS UNIX System Services (USS) operates in one of two different security modes. If the resource profile BPX.DAEMON is not defined, USS is operating in UNIX-level security mode; otherwise, USS is operating in z/OS UNIX security mode.

UNIX-level security provides few options to control access to USS services. A user account that requires access to privileged services must be defined with a UID value of 0, which is referred to as superuser.

z/OS UNIX security provides better access controls to USS services using a SAF security package, such as RACF. A user account can be defined with a non-zero UID and granted permissions to specific resource profiles that protect USS services. Superuser access is granted not with UID 0 but with READ access to the resource BPX.SUPERUSER in the FACILITY class.

Universal Broker

The Universal Broker started task provides services for local and remote Universal Agent managers, such as Universal Command managers or Universal Data Mover managers. Locally executed managers register with the local Universal Broker for monitoring, configuration data, and event recording. Remotely executed managers communicate with the local Universal Broker over a TCP/IP network connection and request execution of server components in order to process local services, such as execution of commands or transfer of data.

Server components initiated by the Universal Broker started task execute as child processes of the Universal Broker process. As such, the server components inherit the user identifier of the Universal Broker process. Some server components, such as Universal Command server, can switch the user identifier with which the work is executed. Switching a user identifier is a privileged operation. The Universal Broker user profile requires sufficient security access for itself and the server components to perform their services.

Universal Broker User and Group Profiles

The security requirements of the Universal Broker depend upon which services are being utilized. By default, all services are configured to be used. Some services can be disabled to reduce the amount of authority the Universal Broker user profile requires. The following table lists the USS privileged services for each component and how to disable the service so that security access to the service is not required.

Service	Description	Disabling
Change directory ownership	Universal Broker dynamically mounts its USS file systems. Once the file systems are mounted, the Broker will initialize them. Initialization consists of changing the ownership of the file systems root directories to the Broker user identifier.	Initialize the file system ownership manually. The Broker will not dynamically mount the file systems and initialize them if they are already mounted and initialized. Make sure the account used to execute the Universal Broker STC owns the location on which the file system is mounted.
Create external links owned by UID 0	Universal Broker dynamically creates external links on the USS file system to select MVS programs in its STEPLIB ddname allocation. The external links are required for the USS spawn function used by the Universal Broker to execute the components. In addition, the UDM Manager also requires external links to execute UCMD Manager or the Universal Connector for SAP (USAP). The external links must be owned by UID 0 when they link to an MVS program that resides in an APF authorized library. Creation of the external links so that they are owned by UID 0 requires superuser access.	Create the external links manually at a permanent location in the USS file system. The content/name of the external links is user-defined. The external links must be owned by UID 0. Specify the external link absolute path name in the component definition START_COMMAND option for UAGSRV, UCMSRV, UCTSRV, and UDMSRV, located in UNVCOMP (UAGCMP00), UNVCOMP(UCSCMP00), UNVCOMP(UTSCMP00), and UNVCOMP(UDSCMP00), respectively. Absolute paths to the external links that Universal Broker creates for UCMD and USAP (for the UDM exec and execsap commands) and UCTL can be specified with the ucmd_path, usap_path, and uctl_path configuration options defined in UNVCONF (UBRCFG00).
Switch user ID and group associations	Universal Command, Universal Data Mover, and Universal Control switch their user IDs with which a work request is executed. The user ID is first authenticated before switching unless there is a Universal Access Control List (UACL) entry that turns authentication off for the request.	Set the Universal Command, Universal Data Mover, and Universal Control servers configuration SECURITY option to a value of NONE. With SECURITY set to NONE, all work requests are executed with the Universal Broker's user ID. Starting with Universal Agent 6.5.0.0, if a valid account password is provided, security contexts can be switched even when the Broker is running without access to BPX.SUPERUSER or BPX.DAEMON. This feature allows the SECURITY option to be set to a value other than NONE, but does not support UACL entries that turn authentication off for a particular request via the 'noauth' parameter. With this option, supplemental group information for the user account will be set, but the process' effective group ID will not be changed. This means that access to system resources must be granted to the specific user account.
Change server component job name	Universal Broker will set the job name of child server processes to the appropriate component name. For example, when the Universal Broker starts a Universal Command server component, the job name is set to UCMSRV.	There is no product configuration option to disable this. By simply not permitting the Broker to the resource profile that protects it, all server components will run with the Universal Broker job name with a numeric value appended to it (for example, UBROKER2).

How to configure the Universal Broker started task user profile to meet security requirements depends on whether USS is running with UNIX-level security or z/OS UNIX security. The following sections describe how to configure the Universal Broker user profile to perform the privileged services listed above for both USS UNIX-level security and z/OS UNIX security configurations.

UNIX-level Security

UNIX-level security refers to a USS security environment where the resource profile BPX.DAEMON is not defined to the FACILITY class.

The only method of permitting a user profile access to privileged services is to define the user profile with a UID value of 0 (superuser). The Universal Broker user profile must be defined with UID 0 to perform any privileged service.

The following steps define the Universal Broker user profile for a UNIX-level security environment:

Step 1	Add the Universal Broker group profile **UBRGRP** using the following RACF command: `ADDGROUP UBRGRP OWNER(SYS1) OMVS(GID(5001))` Change the GID value 5001 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.
Step 2	Add the Universal Broker user profile **UBRUSR** using the following RACF command: `ADDUSER UBRUSR DFLTGRP(UBRGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(0))`

z/OS UNIX Security

z/OS UNIX security refers to a USS security environment where the resource profile BPX.DAEMON is defined to the FACILITY class.

The Universal Broker user profile must be defined with a valid OMVS segment with a non-zero, unique UID value. The user profile security requirements are listed in the following table for each privileged service.

Service	Requirement
Change directory ownership	READ access to BPX.SUPERUSER resource profile in the FACILITY class.
Create external links owned by UID 0	READ access to BPX.SUPERUSER resource profile in the FACILITY class.
Switch user ID and group associations	READ access to BPX.DAEMON and BPX.SUPERUSER resource profiles in the FACILITY class.
Change server component job name	READ access to the BPX.JOBNAME profile in the FACILITY class.

The security requirements can be lifted if the feature that utilizes the service is disabled as described above.

The following steps configure the Universal Broker user profile for a z/OS UNIX security environment:

Step 1	Add the Universal Broker group profile **UBRGRP** using the following RACF command: `ADDGROUP UBRGRP OWNER(SYS1) OMVS(GID(5001))` Change the GID value 5001 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.
Step 2	Add the Universal Broker user profile **UBRUSR** using the following RACF command: `ADDUSER UBRUSR DFLTGRP(UBRGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(5001))` Change the UID value 5001 to a value suitable for your local USS environment. The value must be unique among all user profiles.

Step 3	Permit the Universal Broker user profile READ access to the resource profiles required for enabled services. The following RACF commands permit the user profile to the resources required for all privileged services:
	`PE BPX.DAEMON CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)`
	`PE BPX.SUPERUSER CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)`
	`PE BPX.JOBNAME CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)`
	`SETR RACLIST(FACILITY) REFRESH`

As described above, services can be disabled to make access to BPX.DAEMON and BPX.SUPERUSER optional. By not granting the Universal Broker account access to these resources, the Broker STC will run in unprivileged mode. Access to resources the Broker requires will need to be manually configured, but user context switching is still possible.

Universal Broker Data Access

The Universal Broker user profile **UBRUSR** requires the following access to the data sets allocated in the Universal Broker started task, **UBROKER**:

Ddname	Access	Description
STEPLIB	READ	Program library
UNVCONF	ALTER	Product configuration data
UNVRFC	READ	Universal Connector SAP RFC file
UNVCOMP	ALTER	Product component definition data
UNVNLS	READ	Product national language support data
UNVCREF	READ	Universal Command command reference definitions
UNVDB	UPDATE	Universal Broker HFS component database
UNVSPOOL	UPDATE	Universal Broker HFS spool database

UNVTMPL	READ	Universal Broker configuration templates
UNVTRACE	UPDATE	Universal Broker application trace PDSE
UNVTRMDL	ALTER	Universal Broker application model trace data set
UNVLOG	UPDATE	Universal Broker log data set
UNVKSTR	UPDATE	Universal Broker Keystore data set
UNVAGMDL	ALTER	Universal Automation Center Agent (UAG) model log data set

Universal Enterprise Controller

The Universal Enterprise Controller started task provides services for monitoring and administering Universal Agents distributed throughout the computer network. GUI clients connect to Universal Enterprise Controller to perform tasks and view component activity and statuses.

Universal Enterprise Controller User and Group Profiles

The security requirements of the Universal Enterprise Controller depend upon which services are being utilized. By default, all services are configured to be used. Some services can be disabled to reduce the amount of authority the Universal Enterprise Controller user profile requires. The following table lists the USS privileged services and how to disable the service so that security access to the service is not required.

Service	Description	Disabling
Mount file system	Universal Enterprise Controller dynamically mounts its USS file system. Mounting a file system requires APF authorization or superuser access. Universal Enterprise Controller is not APF authorized.	Statically mount the Universal Enterprise Controller file system. Make sure the account used to execute the Universal Enterprise Controller STC owns the location on which the file system is mounted.
Change directory ownership	Universal Enterprise Controller will initialize its file system if it detects initialization has not been completed. Initialization consists of changing the ownership of the file system root directory to the Universal Enterprise Controller user identifier.	Initialize the file system ownership manually.

How to configure the Universal Enterprise Controller started task user profile to meet security requirements depend on whether USS is running with UNIX-level security or z/OS UNIX security. The following sections describe how to configure the Universal Enterprise Controller user profile to perform the privileged services listed above for both USS UNIX-level security and z/OS UNIX security configurations.

UNIX-level Security

UNIX-level security refers to a USS security environment where the resource profile BPX.DAEMON is not defined to the FACILITY class.

The only method of permitting a user profile access to privileged services is to define the user profile with a UID value of 0 (superuser). The Universal Enterprise Controller user profile must be defined with UID 0 to perform any privileged service.

The following steps define the Universal Enterprise Controller user profile for a UNIX-level security environment:

Step 1	Add the Universal Enterprise Controller group profile **UECGRP** using the following RACF command: `ADDGROUP UECGRP OWNER(SYS1) OMVS(GID(5002))` Change the GID value 5002 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.
Step 2	Add the Universal Enterprise Controller user profile **UECUSR** using the following RACF command: `ADDUSER UECUSR DFLTGRP(UECGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(0))`

z/OS UNIX Security

z/OS UNIX security refers to a USS security environment where the resource profile BPX.DAEMON is defined to the FACILITY class.

The Universal Enterprise Controller user profile must be defined with a valid OMVS segment with a non-zero, unique UID value. The user profile security requirements are listed in the following table for each privileged service.

Service	Requirement
Mount file system	READ access to BPX.SUPERUSER resource profile in the FACILITY class.
Change directory ownership	READ access to BPX.SUPERUSER resource profile in the FACILITY class.

The security requirements can be lifted if the feature that utilizes the service is disabled as described above.

The following steps configure the Universal Enterprise Controller user profile for a z/OS UNIX security environment:

Step 1	Add the Universal Enterprise Controller group profile **UECGRP** using the following RACF command: `ADDGROUP UECGRP OWNER(SYS1) OMVS(GID(5002))` Change the GID value 5002 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.
Step 2	Add the Universal Enterprise Controller user profile **UECUSR** using the following RACF command: `ADDUSER UECUSR DFLTGRP(UECGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(5002))` Change the UID value 5002 to a value suitable for your local USS environment. The value must be unique among all user profiles.

Step 3	Permit the Universal Enterprise Controller user profile READ access to the resource profiles required for enabled services. The following RACF commands permit the user profile to the resources required for all privileged services:
	
	`PE BPX.SUPERUSER CLASS(FACILITY) ID(UECUSR) ACCESS(READ)`
	`SETR RACLIST(FACILITY) REFRESH`

Universal Enterprise Controller Data Access

The Universal Enterprise Controller user profile **UECUSR** requires the following access to the data sets allocated in the Universal Enterprise Controller started task, **UECTLR**:

Ddname	Access	Description
STEPLIB	READ	Program library
UNVCONF	READ	Product configuration data
UNVNLS	READ	Product national language support data
UNVDB	UPDATE	Universal Enterprise Controller HFS database
UNVMSGS	UPDATE	Universal Enterprise Controller message trace data
UNVPRSR	UPDATE	Universal Enterprise Controller parser trace data
UNVTRACE	UPDATE	Universal Enterprise Controller application trace data

Associate Started Tasks with User and Group Profiles

The started tasks must be associated with their user and group profiles defined above. IBM provides two different methods to accomplish this using RACF:

1. STARTED Class Profile
2. Started Procedures Table

Both methods are described below. Only one, not both, of the methods is required. They are provided as examples for your reference. Your local security procedures and processes should be followed.

STARTED Class Profile

The following procedure describes how to associate a user and group profile with the started procedures using the RACF class STARTED method.

Step 1	Define a STARTED class profile for the Universal Enterprise Controller started procedure with the following TSO command: `RDEFINE STARTED UECTLR.* STDATA(USER(UECUSR) GROUP(UECGRP))` The started procedure member name used in the above command is **UECTLR**. If this has been change, the name in the REDEFINE command must also be changed to match.
Step 2	Define a STARTED class profile for the Universal Broker started procedure with the following TSO command: `RDEFINE STARTED UBROKER.* STDATA(USER(UBRUSR) GROUP(UBRGRP))` The started procedure member name used in the above command is **UBROKER**. If this has been change, the name in the REDEFINE command must also be changed to match.
Step 3	The STARTED class must be refreshed to recognize the new profile definitions. The following command assumes that the STARTED class is active and RACLIST'ed. `SETROPTS RACLIST(STARTED) REFRESH`

Started Procedures Table

This section describes how to associate a user and group profile with the started procedures using the RACF started procedure table ICHRIN03 method.

The ICHRIN03 table resides in a system LPA library, such as **SYS1.LPALIB**. Changes to the table require a system IPL using the CLPA option for them to take effect. RACF loads the table at IPL.

Step 1	Add the following entry to the **ICHRIN03** table. The table is an assembly language program that is assembled and link edited into a system LPA library. ``` DC CL8'UECTLR ' PROC NAME DC CL8'UECUSR ' UEC USER PROFILE DC CL8'UECGRP ' UEC GROUP PROFILE DC XL1'00' DC XL7'00' DC CL8'UBROKER ' PROC NAME DC CL8'UBRUSR ' UBR USER PROFILE DC CL8'UBRGRP ' UBR GROUP PROFILE DC XL1'00' DC XL7'00' ```
Step 2	Increment the table count field by two. (The count field is the first 2 bytes of the table.)
Step 3	Assemble and link edit the ICHRIN03 table. IBM provides a sample ICHRIN03 table and the JCL to assemble and link edit it in **SYS1.SAMPLIB(RACTABLE)**.
Step 4	IPL the system with the CLPA option.

z/OS Configuration - Load Library

- Overview
- APF Authorization
- Program Control
- LNKLST

Overview

All Universal Agent programs are installed into the SUNVLOAD PDSE library. The SUNVLOAD library installation requirements are

- Must be APF authorized.
- Must be defined to RACF Program Control.
- Specific programs or SUNVLOAD must be added to the LNKLST only if Universal Automation Center Agent (UAG) is utilized.

The following sections describe the z/OS configuration steps to satisfy these requirements.

APF Authorization

The SUNVLOAD library can be APF authorized using one of the following methods:

- The SETPROG APF console command. The library remains APF authorized only until the next IPL. The SETPROG APF console command is documented in the IBM MVS System Commands manual.
- A PROGxx member of PARMLIB referenced by a SET PROG=xx console command or a PROG=xx statement in IEASYSxx PARMLIB member. The PROGxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual, and the SET console command is documented in the IBM MVS System Commands manual.
- Updating the IEAAPFxx PARMLIB member and perform an IPL. The IEAAPFxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual.

SETPROG APF Command

The SETPROG command temporarily adds a library to the APF list. Use one of the other methods to add the load library permanently to the APF list.

The SETPROG APF console command to add data set #HLQ.SUNVLOAD on volume #SMPVOL to the APF list is illustrated below. Change #HLQ to the appropriate high-level qualifier for your installation and #SMPVOL to the volume on which the library is allocated. SUNVLOAD APF authorization should be permanently established based on local site requirements.

```
SETPROG APF,ADD,DSNAME=#HLQ.SUNVLOAD,VOLUME=#SMPVOL
```

PROGxx PARMLIB Member

The data set can be added permanently to the APF list using the appropriate PROGxx PARMLIB member. The appropriate PROGxx member is referenced in the IEASYSxx PARMLIB member. The PROGxx member can be activated dynamically with the SET PROG=xx console command.

The APF statement of the PROGxx PARMLIB member is illustrated below. The APF statement adds data set #HLQ.SUNVLOAD on volume #SMPVOL to the APF list. Change #HLQ to the appropriate high-level qualifier for your installation and #SMPVOL to the volume on which the library is allocated.

```
APF ADD DSNAME(#HLQ.SUNVLOAD) VOLUME(#SMPVOL)
```

Program Control

Specific services of the z/OS UNIX environment require all programs loaded in the address space to be defined in the security product as controlled. RACF provides program and library control features. By defining a program as controlled, you are implying a certain level of trust, as opposed to a program being run that is not defined as controlled.

The following table identifies the programs in the SUNVLOAD library that must be defined as program controlled.

Program	Description
UCMSRV	Universal Command Server
UDMSRV	Universal Data Mover Server
UCTSRV	Universal Control Server

UECTLR	Universal Enterprise Controller
UAGSRV	Universal Automation Center Agent

The requirement for program control for each of the above programs is the same:

1. Use of the **_passwd** C function to authenticate user ID's
2. Use of the **setuid** C function to switch to the authenticated user ID's

Defining a program or library to RACF program control is accomplished by defining the library to the PROGRAM class.

The following RACF TSO command illustrates defining the library to the PROGRAM class. Change #HLQ to the appropriate high-level qualifier for your installation.

```
RDEF PROGRAM ** ADDMEM('#HLQ.SUNVLOAD'//NOPADCHK) UACC(READ)
```

After defining the library as RACF program controlled, the PROGRAM class must be refreshed. The following RACF TSO command illustrates:

```
SETROPTS WHEN(PROGRAM) REFRESH
```

When all programs executing in an address space are program controlled, the address space is considered a clean environment. All programs loaded dynamically into a clean environment must also be program controlled else the address space is terminated.

Some Universal Agent components may load some IBM modules at runtime to perform requested services. All IBM modules loaded dynamically are loaded from SYS1.LINKLIB. Each member or all members of SYS1.LINKLIB must be set up as program controlled.

The following modules are loaded from SYS1.LINKLIB:

- IGGCSI00 (Catalog Search Interface)
- IEBCOPY (PDS/E copy utility)

SYS1.LINKLIB normally is set up so that all members are program controlled. Verify how your installation has defined SYS1.LINKLIB in regards to the program control facility. If necessary, define the individual members listed above, or all members of SYS1.LINKLIB, as program controlled.

LNKLST

The following Universal Automation Center Agent programs and their aliases must be added to the LNKLST for them to be available to all batch jobs and TSO users:

- UAGRERUN provides batch job rerun capability. The program must be available to all jobs managed by Automation Center.
- UAGCMDZ (and its alias OPSCMDZ) provides a TSO command interface to Universal Controller. The TSO command interface is optional.

The programs are located in the SUNVLOAD library.

If the Universal Automation Center Agent is not used in your installation, this z/OS configuration step can be skipped.

The UAGRERUN and UAGCMDZ programs and their aliases should be added to the LNKLST using one of the following methods:

- Copy the programs and their aliases to a PDSE library that is already in the LNKLST.
- A PROGxx member of PARMLIB referenced by a SET PROG=xx console command or a PROG=xx statement in IEASYSxx PARMLIB member. The PROGxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual, and the SET console command is documented in the IBM MVS System Commands manual.
- Updating the LNKLSTxx PARMLIB member and perform an IPL. The LNKLSTxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual.

Copying Programs and Aliases

The UAGRERUN and UAGCMDZ and their aliases may be copied to a PDSE data set that is already in the LNKLST.

The SUNVLOAD library is a PDSE and the programs are program objects using program management features that are not supported in a PDS. For this reason, the programs must be copied to a PDSE in the LNKLST and not to a PDS.

The JCL below executes IEBCOPY to copy the programs and their aliases to a LNKLST library. Change #HLQ to the appropriate high-level qualifier for your installation and change the data set allocated to ddname OUT to the target PDSE in the LNKLST. The job must end with return code 0.

```
//jobname  JOB CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,COND=(0,NE)
//STEP1    EXEC PGM=IEBCOPY,PARM='RC4NOREP'
//IN       DD  DISP=SHR,DSN=#HLQ.SUNVLOAD
//OUT      DD  DISP=SHR,DSN=lnklst.pdse
//SYSUT3   DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4   DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  COPYGRP I=((IN,R)),O=OUT
  S M=UAGRERUN
  S M=UAGCMDZ
/*
```

SETPROG LNKLST Command

The SETPROG LNKLST command dynamically modifies the LNKLST. The LNKLST changes remain in effect until the next IPL. Use one of the other methods to add the load library permanently to the LNKLST.

LNKLST's are managed as sets. A LNKLST set has a name and consists of an order number of data sets or libraries. To dynamically change the current LNKLST, a LNKLST set must be created, updated, and activated. A newly activated LNKLST set will be active for new address spaces, but existing address spaces are not updated.

The SETPROG LNKLST console command is used to create, update, and activate a LNKLST set. The SETPROG commands to add a data set to the LNKLST are illustrated below. The commands define a LNKLST set name *lnklstname* based on the currently active LNKLST set. It then adds the SUNVLOAD library at the top of the data set list. Lastly, it activates the new LNKLST set. Change #HLQ to the appropriate high-level qualifier for your installation and the LNKLST set name to meet your local installation requirements. Ensure each command completes successfully before executing the next.

```
SETPROG LNKLST,DEFINE,NAME=lnklstname,COPYFROM=CURRENT
SETPROG LNKLST,ADD,NAME=lnklstname,DSNAME=#HLQ.SUNVLOAD,ATTOP
SETPROG LNKLST,ACTIVATE,NAME=lnklstname
```

PROGxx PARMLIB Member

The SUNVLOAD library can be added permanently to the LNKLST using the appropriate PROGxx PARMLIB member. The appropriate PROGxx member is referenced in the IEASYSxx PARMLIB member.

The LNKLST statement of the PROGxx PARMLIB member is illustrated below. The LNKLST statement adds data set #HLQ.SUNVLOAD to the LNKLST set *lnklstname*. Change #HLQ to the appropriate high-level qualifier for your installation and *lnklstname* to the appropriate LNKLST set name for your installation.

```
LNKLST ADD NAME(lnklstname) DSNAME(#HLQ.SUNVLOAD)
```

z/OS Configuration - SMF Exits

- Overview
- SMF Exit Introduction
- Universal Command Server
  - Configuring SMF
  - Installing SMF Exit Routines
- Universal Automation Center Agent
  - Configuring SMF
  - Installing SMF Exit Routines

Overview

SMF exits are utilized by the following Universal Agent components to provide their z/OS services:

- Universal Command Server uses one SMF exit to provide support for starting Started Tasks (STC) using Universal Command Manager. If Universal Command STC support is not required, the SMF exit does not need to be installed.
- Universal Automation Center Agent uses a number of SMF exits to provide a range of services available in Universal Controller.

The following sections describe how to install the SMF exits for each component. The SMF exits are only required if the component or component services are required.

SMF Exit Introduction

MVS System Management Facilities (SMF) collects and records system events in the form of SMF records. An SMF record is identified with a numeric record type. In addition to recording system events, SMF provides exit points in the control program from which system or application provided exit routines or programs can be called at the time work is being processed. The SMF exit routines can take appropriate actions based on the point at which the exit is called and the information provided in the SMF record. The SMF exits allow for systems and applications to monitor and augment the processing of MVS workload. For additional information on SMF, refer to the IBM manuals *MVS System Management Facilities (SMF)*, *MVS Installation Exits*, and *MVS Initialization and Tuning Reference*.

SMF configuration is performed with PARMLIB member SMFPRMxx. SMF configuration relevant for product installation is the SMF record types being recorded and the SMF exits defined. The SMFPRMxx parameters SYS and SUBSYS define these SMF recording options. The SYS parameter specifies system wide recording options for all subsystems (JES2, TSO, STC, etc.). The SUBSYS parameter specifies subsystem specific recording options. The SUBSYS parameter options override the equivalent options specified on the SYS parameter. Any SUBSYS parameter option not specified will default to the SYS parameter option.

The following example illustrates how SMF SYS and SUBSYS parameters work together.

```
SYS( EXITS(IEFU83,IEFU84) )
SUBSYS( STC, EXITS(IEFU83) )
```

The SYS parameter EXITS option defines the two SMF exits IEFU83 and IEFU84. The SUBSYS parameter for the STC subsystem also specifies the EXITS option, overriding the SYS EXITS option. The SUBSYS parameter EXIT option defines one SMF exit IEFU83 for the STC subsystem. Consequentially, only the IEFU83 exit is called for STC subsystem workload. The IEFU84 exit is not called for STC subsystem workload. For all workload other then the STC subsystem, the SMF exits IEFU83 and IEFU84 are called.

Once SMFPRMxx has defined the appropriate SMF record types to collect and the SMF exits to activate, the product SMF exit routines are installed. IBM provides the dynamic exit facility to add and remove exit routines dynamically and at IPL. IBM has defined all the SMF exits to the dynamic exit facility. The exit routines are added or removed from SMF exits using the PROGxx PARMLIB members. The PROGxx EXIT statements used to add product exit routines are listed in the appropriate sections below.

The exit names used in the dynamic exit facility are formatted as `SYSssn.exitname`, where `ssn` is the subsystem name, or blank if it is a system wide exit, and `exitname` is the name of the exit, such as IEFU83. The following table provides some example dynamic exit facility exit names.

Name	Description
`SYS.IEFU83`	The default SMF exit IEFU83 for all system workload types.
`SYSSTC.IEFU83`	The SMF exit IEFU83 for STC subsystem workload.
`SYSJES2.IEFU83`	The SMF exit IEFU83 for JES2 subsystem workload.

Note

When using SMF Exit items, please check against conflicting existing SMF definitions, such as `SYSJES2.IEFXXX`.

Universal Command Server

The Universal Broker STC establishes the environment to support STC execution by Universal Command Server. Part of the environment consists of adding SMF exit routine UNVACTRT to SMF exit point SYSSTC.IEFACTRT.

Note

The UNVACTRT exit should reside in LPA, the LNKLST concatenation, or the nucleus. Do not use the DSNAME keyword when defining the exit in PROGxx; the system will not be able to load the exit when restarting SMF.

The Broker's UCMD_STC_SUPPORT option specifies whether or not the environment for STC support should be established:

- **yes** (the default) specifies that the environment should be established.
- **no** specifies that the environment should not be established.

Configuring SMF

Configuring SMF consists of defining SMF exit point SYSSTC.IEFACTRT for the STC subsystem.

The exit point is defined in the SMFPRMxx PARMLIB member with the SUBSYS STC EXITS parameter either implicitly, by excluding the EXITS parameter, or explicitly.

The following SUBSYS statement illustrates how to define SYSSTC.IEFACTRT:

```
SUBSYS(STC,EXITS(IEFACTRT,...))
```

Once the SMFPRMxx PARMLIB member has been modified, the SMFPRMxx member can be activated dynamically by restarting SMF with the following command:

```
SET SMF=xx
```

The following console command displays the active SMF options:

```
D SMF,O
```

Installing SMF Exit Routines

The SMF exit routine UNVACTRT is provided as part of the Universal Agent package. UNVACTRT must be added to SMF exit point SYSSTC.IEFACTRT. The exit routine can be added automatically by the Broker STC when it starts or statically in a PARMLIB member.

Automatic Installation

The Broker dynamically adds SMF exit routine UNVACTRT to the SYSSTC.IEFACTRT exit point if the SMF_EXIT_LOAD_LIBRARY configuration option is defined in the Universal Broker configuration file. The option indicates that the Broker dynamically add UNVACTRT and specifies the load library from which UNVACTRT is loaded.

Static Installation

If the Universal Broker configuration option SMF_EXIT_LOAD_LIBRARY is not specified in the Universal Broker configuration file, Universal Broker assumes that exit routine UNVACTRT already is added and will not attempt to add it when it starts.

Exit routine UNVACTRT is permanently added to SYSSTC.IEFACTRT using the appropriate PROGxx PARMLIB member. The EXIT statement of the PROGxx PARMLIB member is illustrated below. The EXIT statement adds exit routine UNVACTRT from the #HLQ.SUNVLOAD load library to the SMF exit point SYSSTC.IEFACTRT. Change #HLQ to the appropriate high-level qualifier for your installation. The exit is no longer given control if it encounters an ABEND.

```
EXIT ADD EXITNAME(SYSSTC.IEFACTRT) MODNAME(UNVACTRT) ABENDNUM(1)
```

The UNVACTRT exit routine is added when the system is IPL'ed and the PROGxx member is processed. The SMF exit can be added dynamically without an IPL by activating PROGxx member with the SET console command. The following SET command activates PROGxx member:

```
SET PROG=xx
```

The PROGxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual. The SET console command is documented in the IBM MVS System Commands manual.

The following console command displays the exit routines installed for exit point SYSSTC.IEFACTRT:

```
D PROG,EXIT,EN=SYSSTC.IEFACTRT,DIAG
```

Universal Automation Center Agent

The Universal Automation Center Agent (UAG) establishes the environment to support Universal Controller services on z/OS. Part of the environment consists of establishing the following exit routines:

Exit Point	Exit Routine
CNZ_WTOMDBEXIT	UAGWMDBX
SYS.IEFUSI	UAGUSI
SYS.IEFU83	UAGU83
SYS.IEFU84	UAGU84
SYSSTC.IEFU83	UAGU83

Note

When using SMF Exit items, please check against conflicting existing SMF definitions, such as `SYSJES2.IEFXXX`.

Note

UAG versions below 6.6.0.0 require the UAGUJV exit.

UAG versions below 6.4.3.0 require the UAGUJI exit.

If multiple versions of UAG are running on a single z/OS system, older versions of UAGUJV and UAGUJI can be used alongside newer versions of IEFUSI, IEFU83 and IEFU84. UAGUJV and UAGUJI will be ignored by newer versions of UAG that don't need them, and newer versions of IEFUSI, IEFU83 and IEFU84 will work appropriately with older versions of UAG.

Configuring SMF

Configuring SMF consists of specifying required SMF record types to collect and the required exit points to define.

The SMF configuration must meet the following requirements:

- SMF record types 14, 15, 16, 17, 18, 30, and 70 must be collected.
- System exit points IEFUSI, IEFU83, and IEFU84 must be defined.
- Assuming the Universal Command SMF exit is installed or there is already a SUBSYS EXITS option for STC, the subsystem STC exit point IEFU83 must be defined.

The record types and exit points are defined in the SMFPRMxx PARMLIB member. The following SMFPRMxx statements illustrate how to define the record types to collect and exit points.

The SYS statement uses the NOTYPE parameter to exclude specific record types. This is one of many ways the SYS statement can be specified to meet the SMF record type requirements.

```
SYS(NOTYPE(19,62:69,99),EXITS(IEFU83,IEFU84,IEFUSI,...))
SUBSYS(STC,EXITS(IEFU83,...))
```

Once the SMFPRMxx PARMLIB member has been modified, the SMFPRMxx member can be activated dynamically by restarting SMF with the following command:

```
SET SMF=xx
```

The following console command displays the active SMF options:

```
D SMF,O
```

Installing SMF Exit Routines

The SMF exit routines are installed using the appropriate PROGxx PARMLIB member. The EXIT statements for the PROGxx PARMLIB member are illustrated below. The EXIT statements add all the exit routines from the #HLQ.SUNVLOAD load library to the appropriate exit points. Change #HLQ to the appropriate high-level qualifier for your installation.

```
EXIT ADD EXITNAME(SYS.IEFUSI) MODNAME(UAGUSI) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYS.IEFU83) MODNAME(UAGU83) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYS.IEFU84) MODNAME(UAGU84) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYSSTC.IEFU83) MODNAME(UAGU83) DSNAME(#HLQ.SUNVLOAD)
```

CNZ_WTOMDBEXIT is not an SMF exit, so it can be configured simply by coding an EXIT statement in a PROGxx PARMLIB member.

For example:

```
EXIT ADD EXITNAME(CNZ_WTOMDBEXIT) MODNAME(UAGWMDBX) DSNAME(#HLQ.SUNVLOAD)
```

The exit routines are added when the system is IPL'ed and the PROGxx member is processed. The SMF exit routines can be added dynamically without an IPL by activating PROGxx member with the SET console command. The following SET command activates PROGxx member.

```
SET PROG=xx
```

The PROGxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual. The SET console command is documented in the IBM MVS System Commands manual.

z/OS Configuration - JES SYSOUT Processing

- Overview
- Configuration
  - Multiple SYSOUT Applications
  - UAG SYSOUT Management
  - Default Configuration

Note

The information in this section applies to UAG release 5.1.0.12 and above.

Overview

This page describes how to configure Universal Automation Center Agent (UAG) to reliably process JES SYSOUT data sets. If UAG is not used, this information can be skipped.

The batch jobs that UAG submits create JES SYSOUT data sets. UAG processes the following SYSOUT data sets produced by the jobs:

- UAGRERUN report produced by the first step of every batch job UAG submits. The SYSOUT data set is written by step name OPSSTP00 to ddname SYSPRINT.
- JES system data sets produced by every batch job. The JES system data sets include the JESMSGLG, JESJCL and JESYSMSG ddnames.

UAG processes the SYSOUT data sets for the following purposes:

1. Retrieving the UAGRERUN report for storage in the Universal Controller.
2. Analyzing JCL errors that occur during JCL conversion or at run-time.
3. Optionally, retrieving the JES system data sets for storage in the Universal Controller.

JES SYSOUT data sets cannot be processed by more then one application at time. Applications, such as spool archivers and JES offloaders, must process SYSOUT data sets one at a time. If multiple applications process SYSOUT data sets simultaneously, some SYSOUT data sets potentially will not be processed.

For UAG to reliably process SYSOUT data sets, a JES held class must be dedicated exclusively to UAG. A JES class is defined as held with a JES JOBCLASS statement with an OUTDISP parameter value of HOLD,HOLD.

Configuration

UAG provides a flexible SYSOUT processing configuration to accommodate a variety of environments. The best configuration depends on your existing SYSOUT management practices and processes.

There are three UAG configuration options for SYSOUT processing:

- JES_SYSOUT_CLASS specifies the JES held class dedicated to UAG SYSOUT processing.
- JES_SYSOUT_DISP specifies the disposition of the SYSOUT data sets once UAG has completed SYSOUT processing.
- JES_SYSOUT_RETENTION specifies the number of hours that job SYSOUT files are retained in the UAG cache directory.

The following sections discuss some possible UAG SYSOUT configurations. The SYSOUT options allow UAG SYSOUT processing to be customized to meet almost any SYSOUT management requirement. The following configurations are the most common.

Multiple SYSOUT Applications

This configuration addresses environments that have existing SYSOUT applications that process job SYSOUT data sets. These applications are often called - for example - spool archivers, output management systems, or job log archivers. They typically read all SYSOUT data sets from one or more JES held classes and archive the SYSOUT in a database. Once they have finished processing the SYSOUT in the JES spool, the SYSOUT is typically deleted from the spool.

UAG must be properly configured to work with other SYSOUT processing applications. Since SYSOUT data sets cannot be processed simultaneously by multiple SYSOUT applications, each application must process the SYSOUT data sets one at a time.

The first step is to select a JES held class to dedicate to UAG. The class cannot be shared with any other SYSOUT application. Specify the class with the UAG JES_SYSOUT_CLASS configuration option. By specifying a JES_SYSOUT_CLASS value, UAG will modify the JOB statement of the jobs it submits to JES. The JOB statement MSGCLASS parameter is changed to the JES_SYSOUT_CLASS value. If no MSGCLASS parameter is present on the JOB statement, it will be added. The JOB statement MSGCLASS parameter specifies the JES class where the job's JES system data sets are spooled.

Once UAG has completed processing the SYSOUT data sets, it performs final disposition on them as specified by the UAG JES_SYSOUT_DISP configuration option. The SYSOUT data sets must be moved to the JES class or classes used by the next SYSOUT application to process them.

There are two ways to specify the JES class to which to move the SYSOUT data sets:

- If the original JOB statement MSGCLASS value correctly specified the class, then a JES_SYSOUT_DISP value of **KEEP,*** will keep the SYSOUT data sets and move them to the original MSGCLASS class. This configuration requires no JCL changes nor changes to the other SYSOUT application.
- The class can be specified directly. For example, a JES_SYSOUT_DISP value of **KEEP,H** will keep the SYSOUT data sets and move them to held class H.

UAG SYSOUT Management

UAG z/OS task definitions may be defined to retrieve the job log (that is the job's JES system data sets) once the job completes processing. The job logs are transmitted to the Controller and stored in the database. The job logs can be viewed from the Universal Controller web interface. In this configuration, once UAG has completed retrieving the job log, the SYSOUT data sets can be deleted from the JES spool. It is assumed that there are no other SYSOUT applications processing the SYSOUT data sets other then UAG.

The JES_SYSOUT_CLASS option is not required in this configuration. By default, UAG will process a job's SYSOUT data sets from any JES class in which it finds them.

Once UAG has completed processing the SYSOUT data sets, it performs final disposition on them as specified by the UAG JES_SYSOUT_DISP option. Since UAG is the one and only SYSOUT application, the SYSOUT data sets can be deleted. A JES_SYSOUT_DISP value of **DELETE** will remove the SYSOUT data sets from the JES spool.

Default Configuration

The UAG default configuration is backwardly compatible with previous UAG releases. By default UAG will process SYSOUT data sets in the JES spool regardless of their disposition (held or not) or class. Consequentially, if the SYSOUT data sets are processed by another application, the results are unreliable. Once UAG completes processing of the SYSOUT data sets, it leaves them in the JES spool without changing their disposition or class.

Universal Agent Database Configuration

- z/OS UNIX File System Introduction
- zFS Configuration
- Mounting and Unmounting the Databases
  - Dynamic Mounts
  - Manual Mounts
  - TSO Commands
  - Console Commands
  - BPXPRMxx
  - Data Set Initialization
- Memory Management

z/OS UNIX File System Introduction

The z/OS implementation of Universal Agent databases utilizes the z/OS UNIX File System. Stonebranch recommends the use of the z/OS File System (zFS) for use by the Universal Broker and Universal Enterprise Controller started tasks. In a Sysplex environment, use of zFS is required.

zFS is a file system used by z/OS UNIX System Services (USS). It is a POSIX conforming hierarchical file system stored in one or more zFS data sets bound together into one hierarchical directory structure. A single zFS data set consists of a directory tree and files. Refer to the IBM UNIX System Services Planning manual for a complete discussion of the z/OS UNIX file system and its administration.

A zFS data set must be mounted before a program can access any file or directory within it. A mount operation binds the root directory of the zFS data set to an existing directory in the hierarchical file system referred to as the mount point. After the mount operation completes, the zFS data set's directory structure becomes part of the file system hierarchy starting at the mount point. A zFS data set can only be mounted one at a time.

The mount operation makes the files and directories within the zFS data set accessible to all users. User access is controlled with directory and file permissions contained within the zFS data set. Initially, a zFS data set's root directory is owned by the user that allocated the data set, and the directory permissions are set so that only that user has read, write, and execute permissions (permission mode 700). No other users have access.

zFS Configuration

zFS data sets are created by the installation JCL. The zFS data sets are used by Universal Broker and Universal Enterprise Controller.

A zFS data set is referred to as a zFS aggregate. Universal Agent uses compatibility mode aggregates only.

The Universal Broker zFS data set names must be specified with the UNIX_DB_DATA_SET and UNIX_SPOOL_DATA_SET Universal Broker configuration options. The Universal Enterprise Controller zFS data set name must be specified with the UNIX_DB_DATA_SET Universal Enterprise Controller configuration option.

Note

Unless otherwise stated in the release notes or install instructions, backward compatibility is always preserved in the Universal Broker and Universal Enterprise Controller databases. After completing the steps listed in any of the upgrade scenarios listed here, any existing databases used for the old version also can be used after the upgrade. This means that creating new databases using install job **UNVIN07** is not necessary when upgrading.

Mounting and Unmounting the Databases

When the Universal Broker and Universal Enterprise Controller started tasks are started, they check if their zFS data sets have been mounted. If they are mounted, the started tasks will attempt to use them. If they are not mounted, the started tasks will attempt mount the data sets dynamically.

Note

In a Sysplex, the data set specified on UNIX_SPOOL_DATA_SET must be mounted in RWSHARE mode on a file system that is shared across the Sysplex. The mount point specified on SHARED_MOUNT_POINT must exist on a shared file system.

Dynamic Mounts

The started tasks will mount the zFS data sets if they are not mounted. The data sets are mounted at mount points defined in the directory specified in the Universal Broker configuration options:

- MOUNT_POINT, which defaults to the **/tmp** directory, is used as the location to mount file systems which are not shared in a Sysplex.
- SHARED_MOUNT_POINT, which defaults to the location specified for MOUNT_POINT, is used as the location to mount file systems which are shared in a Sysplex (currently only UNVSPOOL).

In a non-Sysplex environment, SHARED_MOUNT_POINT can be ignored. The mount points are sub-directories named after the data set names. For example, if the zFS data set name is **UNV.UNVDB**, the mount point is **/tmp/UNV.UNVDB**.

When the started tasks mount a zFS data set, the mount parameter **AGGRGROW** is used to specify that the zFS data set should automatically utilize secondary extents to expand if it runs out of allocated space.

The zFS data sets are not unmounted when the started tasks are stopped. It is not known whether other users are using the mounted data sets.

Manual Mounts

The started tasks will use the existing mounts of the zFS data sets. Dynamic mounts provide the easiest administration, but you may want to manually mount the data sets to take advantage of several available mount options. For example, the FSFULL PARM value can be used to issue operator messages when a file system reaches a specified percent full.

When mounting zFS data sets, the mount parameter **AGGRGROW** should be used to specify that the zFS data set should automatically utilize secondary extents to expand if it runs out of allocated space.

In a Sysplex, the mount parameter **RWSHARE** should be used when mounting the UNVSPOOL file system.

When the zFS data sets are manually mounted, the mount point can be any z/OS UNIX directory that satisfies the requirements of the file system. The name of the directory does not matter. The started tasks will locate the mount point regardless of location or name.

zFS data sets can be mounted using the TSO MOUNT command, the USS mount command or with PARMLIB member **BPXPRMxx** at IPL. The TSO MOUNT and USS mount commands mount it for the current IPL only while the **BPXPRMxx** member will mount the data set for each IPL.

zFS data sets can be unmounted using the TSO UNMOUNT or USS unmount commands or with the MODIFY BPXOINIT console command.

TSO Commands

The TSO commands to mount and unmount zFS data set **UNV.UNVDB** at mount point **/opt/unvdb** are illustrated below:

zFS Mount Command

```
MOUNT FILESYSTEM('UNV.UNVDB') MOUNTPOINT('/opt/unvdb') TYPE(ZFS) PARM(AGGRGROW)
```

Unmount Command

```
UNMOUNT FILESYSTEM('UNV.UNVDB')
```

The TSO command to mount zFS data set **UNV.UNVSPOOL** at mount point **/shared/unvspool** in a Sysplex is illustrated below:

zFS Mount Command

```
MOUNT FILESYSTEM('UNV.UNVSPOOL') MOUNTPOINT('/shared/unvspool') TYPE(ZFS) PARM('AGGRGROW,RWSHARE')
```

The user ID that issues the mount or unmount commands must have an OMVS UID of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class.

Console Commands

The console commands to unmount zFS data set **UNV.UNVDB** is illustrated below in addition to the console command to list currently mounted data sets.

Unmount Command

```
F BPXOINIT,FILESYS=UNMOUNT,FILESYSTEM=UNV.UNVDB
```

Note

A console reply message will ask for confirmation.

Display Command

```
D OMVS,FILE
```

BPXPRMxx

The BPXPRMxx statement to mount zFS data set **UNV.UNVDB** at mount point **/opt/unvdb** is illustrated below:

```
MOUNT FILESYSTEM('UNV.UNVDB') TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/opt/unvdb') PARM('AGGRGROW')
```

Both zFS data sets must be mounted with mode read/write, which is the default.

In a Sysplex configuration, file system UNVSPOOL should be mounted with PARM('AGGRGROW,RWSHARE').

Data Set Initialization

When the started tasks start, they find the mount point for their zFS data sets. Regardless of whether the zFS data sets were dynamically mounted or statically mounted, the started tasks check for an initialization flag file named **.inited** in the root directory of the mounted data set.

If the file is not found, which is the case when they are first mounted, the started tasks change the owner of the root directory to the user ID with which they are executing and change the permission mode to the MOUNT_POINT_MODE configuration option value, which defaults to 750. In a Sysplex configuration, SHARED_MOUNT_POINT_MODE, which defaults to the value specified on MOUNT_POINT_MODE, can be used to specify a different set of permissions for the shared mount point.

If you want to customize either the owner or permission of the directories, manually create the **.inited** file in the root directory of the zFS data set to prevent the started tasks from performing the initialization when they start. The USS command **touch.inited** can be used to create an empty file.

Memory Management

Berkeley DB uses a temporary cache in memory to manage its databases. If this cache becomes sufficiently large, it must be written to disk.

Berkeley DB has a default location for storing temporary cache files, but if UEC cannot access that location, or there is no space to write these files in the default location, the following error can occur in UEC, and UEC shuts down:

UNV4301D Database error: 'temporary: write failed for page XXXXX'

To work around this issue, the following steps write the temporary cache files to the UEC database directory:

Step 1	Mount the **UECDB** zFS data set.
Step 2	Inside the mount point, create a text file named **DB_CONFIG**.
Step 3	Inside the **DB_CONFIG** file, add the following string: **`set_tmp_dir *dbpath*`** Where **dbpath** is the path to the location in which the database files reside.
Step 4	Start / restart UEC.

z/OS Installation - Customization

- Overview
- Universal Broker Customization
  - Universal Broker Configuration
  - Universal Broker JCL Procedure
- Universal Automation Center Agent Customization
  - Universal Automation Center Agent Configuration
- Universal Certificate Customization
  - Universal Certificate JCL Procedure
- Universal Command Manager Customization
  - Universal Command Manager Configuration
  - Universal Command Manager JCL Procedure
- Universal Command Server Customization
  - Universal Command Server Configuration
- Universal Connector Customization
  - Universal Connector Configuration
  - Universal Connector SAP RFC Configuration
  - Universal Connector JCL Procedure
- Universal Control Manager Customization
  - Universal Control Manager Configuration
  - Universal Control Manager JCL Procedure
- Universal Control Server Customization
  - Universal Control Server Configuration
- Universal Data Mover Manager Customization
  - Universal Data Mover Manager Configuration
  - Universal Data Mover Manager JCL Procedure
- Universal Data Mover Server Customization
  - Universal Data Mover Server Configuration
- Universal Enterprise Controller Customization
  - Universal Enterprise Controller Configuration
  - Universal Enterprise Controller JCL Procedure
- Universal Event Monitor Manager Customization
  - Universal Event Monitor Manager Configuration
  - Universal Event Monitor Manager JCL Procedure
- Universal Query Customization
  - Universal Query Configuration

Overview

This page provides the following information for the customization of Universal Agent components:

- Configuration
- JCL procedure

(For information on applying product licenses to installed Universal Agent for z/OS components, see z/OS Installation - Licensing.)

Universal Broker Customization

Universal Broker Configuration

Universal Broker for z/OS uses a configuration file for its customizations. The configuration file is member **UBRCFG00** in the **UNVCONF** library allocated to the **UNVCONF** ddname in the started procedure's JCL.

See the Universal Broker 7.2.x Reference Guide for details on configuring Universal Broker.

Universal Broker JCL Procedure

A JCL procedure is provided in member **UBROKER** in library **SUNVSAMP**.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information.

See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

Universal Automation Center Agent Customization

Universal Automation Center Agent Configuration

There are two files used in defining the Universal Automation Center Agent (UAG) configuration:

- UAG component definition file
- UAG configuration file

The execution of UAG is managed by Universal Broker. The component definition file defines UAG attributes to the Broker. The Broker uses the component definition file to manage the execution of UAG. The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UAGCMP00**.

The UAG configuration file defines system-wide customizations for UAG features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task. The Broker passes the data set name to UAG when it is started by the Broker. The UAG component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UAGCFG00**.

See the Universal Automation Center Agent 7.2.x Reference Guide for details on configuring UAG.

Universal Certificate Customization

Universal Certificate JCL Procedure

A JCL procedure is provided in member **UCRPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//        JCLLIB ORDER=UNV.SUNVSAMP
//STEP1   EXEC UCRPRC
```

Universal Command Manager Customization

Universal Command Manager Configuration

Universal Command Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UCMCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Manager.

Universal Command Manager JCL Procedure

A JCL procedure is provided in member **UCMDPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//        JCLLIB ORDER=UNV.SUNVSAMP
//STEP1   EXEC UCMDPRC
```

Universal Command Server Customization

Universal Command Server Configuration

There are two files used in defining the Universal Command Server configuration:

- Server's component definition file
- Server's configuration file

The execution of the Server is managed by Universal Broker. The component definition file defines the Server attributes to the Broker. The Broker uses the component definition file to manage the execution of the server. The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UCSCMP00**.

The Universal Command Server configuration file defines system-wide customizations for Server features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task. The Broker passes the data set name to the Server when it is started by the Broker. The Server component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UCSCFG00**.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Server.

Universal Connector Customization

Universal Connector Configuration

Universal Connector for z/OS uses a configuration file for system-wide customizations. The configuration file is member **USPCFG00** in the **UNVCONF** library allocated to ddname **UNVRFC** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the Universal Connector for SAP 7.2.x Reference Guide for details on configuring Universal Connector.

Universal Connector SAP RFC Configuration

Universal Connector utilizes SAP's RFC interface. The RFC interface must be configured to meet your local SAP environment. The RFC configuration file is member **USPRFC00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

Universal Connector JCL Procedure

A JCL procedure is provided in member **USPPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//        JCLLIB ORDER=UNV.SUNVSAMP
//STEP1   EXEC USPPRC
```

Universal Control Manager Customization

Universal Control Manager Configuration

Universal Control Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UCTCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Manager.

Universal Control Manager JCL Procedure

A JCL procedure is provided in member **UCTLPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//        JCLLIB ORDER=UNV.SUNVSAMP
//STEP1   EXEC UCTLPRC
```

Universal Control Server Customization

Universal Control Server Configuration

There are two files used in defining the Universal Control Server configuration:

- Server's component definition file.
- Server's configuration file.

The execution of the Server is managed by Universal Broker. The component definition file defines the Server attributes to the Universal Broker, which uses the file to manage the execution of the Server.

The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UTSCMP00**.

The Universal Control Server configuration file defines system-wide customizations for Server features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task.

The Universal Broker passes the data set name to the Server when it starts the Server. The Server component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UTSCFG00**.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Server.

Universal Data Mover Manager Customization

Universal Data Mover Manager Configuration

Universal Data Mover Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UDMCFG00** in the **UNV CONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring Universal Data Mover.

Universal Data Mover Manager JCL Procedure

A JCL procedure is provided in member **UDMPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//        JCLLIB ORDER=UNV.SUNVSAMP
//STEP1   EXEC UDMPRC
```

Universal Data Mover Server Customization

Universal Data Mover Server Configuration

There are two files used in defining the Universal Data Mover Server configuration:

- Server's component definition file
- Server's configuration file

The execution of the Server is managed by Universal Broker. The component definition file defines the Server attributes to the Universal Broker, which uses the file to manage the execution of the server.

The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UDSCMP00**.

The Universal Data Mover Server configuration file defines system-wide customizations for Server features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task.

The Universal Broker passes the data set name to the Server when it starts the Server. The Server component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UDSCFG00**.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring Universal Data Mover Server.

Universal Enterprise Controller Customization

Universal Enterprise Controller Configuration

Universal Enterprise Controller for z/OS uses a configuration file for its customizations.The configuration file is member **UECCFG00** of the **UNVCONF** library allocated to ddname **UNVCONF** in the started procedure's JCL.

See the Universal Enterprise Controller 7.2.x Reference Guide for details on configuring Universal Data Mover.

Universal Enterprise Controller JCL Procedure

A JCL procedure is provided in member **UECTLR** in library **SUNVSAMP**.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

Universal Event Monitor Manager Customization

Universal Event Monitor Manager Configuration

Universal Event Monitor Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UEMCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the Universal Event Monitor 7.2.x Reference Guide for details on configuring Universal Event Monitor Manager.

Universal Event Monitor Manager JCL Procedure

A JCL procedure is provided in member **UEMPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See z/OS Installation - Time Zone Environment Variable for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the JCLLIB JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//        JCLLIB ORDER=UNV.SUNVSAMP
//STEP1   EXEC UEMPRC
```

Universal Query Customization

Universal Query Configuration

Universal Query for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UQRCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the Universal Agent Utilities 7.2.x Reference Guide for details on Universal Query configuration options.

z/OS Installation - Cumulative PTF Maintenance

- z/OS Installation - Cumulative PTF Maintenance
- Installation Procedures
- Cumulative PTF Maintenance Distribution File
  - Compressed Archive
  - ZIP Archive
- Distribution File Contents
- Transferring a Cumulative PTF File to z/OS
  - Extract the Cumulative PTF Files
  - Transfer the Cumulative PTF Files
  - Receive the Cumulative PTF Datasets
- Installing a Cumulative PTF File

z/OS Installation - Cumulative PTF Maintenance

This page describes the procedures for installing a cumulative PTF maintenance for Universal Agent for z/OS.

Installation Procedures

Installation of a cumulative PTF maintenance is comprised of the following procedures:

Step 1	Download one of the Universal Agent 7.2.x for z/OS cumulative PTF maintenance distribution files. Cumulative PTFs are packaged as a compressed tarball (`sb-v.r.m.l-zos-PTFs.tar.Z`) or as a ZIP file (`sb-v.r.m.l-zos-PTFs.zip`).
Step 2a	If the distribution files with the `.tar.Z` extension was downloaded, decompress it using any utility that can process files in Lempel-Ziv (that is, **compress**) format. Use **tar** to extract the files from the uncompressed file. To do both in a single statement, decompress the file to stdout and pipe the result to tar (for example, `zcat sb-v.r.m.l-zos-PTFs.tar.Z | tar xvf -`).
Step 2b	If the distribution file with the `.zip` extension was downloaded, decompress it using any utility that can process files in **ZIP** (or, PKZIP) format. The `.zip` file can also be extracted using Windows native file extraction tools.
Step 3	Modify the `UNVFTP-PTF.TXT` script provided by the distribution file to upload the PTF files and the JCL (which resides in `UNVRECV-PTF.JCL`) to process them. Two versions of these files are included in the distribution file. The files in the main directory are EBCDIC-encoded files. ASCII-encoded versions of these files are in the `./ascii subdirectory`. Look for the #HOSTNAME, #USERID, #PSWD, and #HLQ placeholders in `UNVFTP-PTF.TXT` and set them to the appropriate values.
Step 4	Modify the JCL uploaded from `UNVRECV-PTF.JCL` and run the job to receive the cumulative PTF XMIT file.

Cumulative PTF Maintenance Distribution File

To obtain a cumulative PTF maintenance, you must download the corresponding z/OS - PTFs package from the Stonebranch Customer Portal.

A customer user name and password — provided by Stonebranch, Inc. — are required to access this area.

Compressed Archive

This distribution file is a compressed TAR file that contains the cumulative PTFs in TSO TRANSMIT format, along with supporting files used to upload and receive the cumulative PTFs.

This file uses the following naming convention:

```
sb-Version.Release.Modification Level.Maintenance Level-operating system-PTFs.tar.Z
```

For example: `sb-7.0.0.1-zos-PTFs.tar.Z`.

The cumulative PTF XMIT dataset can be allocated and uploaded using the sample `UNVFTP-PTF.TXT`, and the sample JCL provided in `UNVRECV-PTF.JCL` can be used to do a TSO RECEIVE on the uploaded dataset.

ZIP Archive

The ZIP version of the z/OS Universal Agent cumulative PTF distribution files have the same contents as their compressed archive counterparts and use the following naming convention:

```
sb-Version.Release.Modification Level.Maintenance Level-operating system-PTFs.zip
```

For example: `sb-7.2.0.1-zos-PTFs.zip`.

To unzip the file, use a third-party ZIP-based extraction application or the native Windows file extraction tool available from Windows Explorer.

For example, to extract the files using Windows File Explorer (or just Windows Explorer in some older Windows versions), right-click on the file name, select Extract All from the pop-up menu, and follow the instructions.

Distribution File Contents

The following table lists the files included in the Universal Agent for z/OS cumulative PTF distribution files.

File Name	Description
README.TXT	Contains instructions for preparing and executing the install. Note This is an EBCDIC-encoded text file.
UNVFTP-PTF.TXT	A sample FTP script that allocates and transfers datasets used to store the distribution file's contents on the z/OS system. Note This is an EBCDIC-encoded text file.
UNVRECV-PTF.JCL	Sample JCL that may be used to execute a TSO RECEIVE on the uploaded UNVVRMxx.XMT file. Note This is an EBCDIC-encoded text file.
UNVVRMxx.XMT	A file in TSO TRANSMIT format that contains cumulative PTF file for the Universal Agent release identified by VRM (version, release, modification). 'xx' is a sequential number starting at 01.
ascii/README.TXT	An ASCII-encoded copy of README.TXT.
ascii/UNVFTP-PTF.TXT	An ASCII-encoded copy of UNVFTP-PTF.TXT.
ascii/UNVRECV-PTF.JCL	An ASCII-encoded copy of UNVRECV-PTF.JCL.

Transferring a Cumulative PTF File to z/OS

After downloading the distribution file to your workstation, you must extract the files and transfer them to the z/OS system upon which Universal Agent for z/OS is installed.

Follow the steps below to transfer the distribution file's contents to the z/OS system.

1. Extract the cumulative PTF file from the distribution file.
2. Transfer the cumulative PTF file.
3. Receive the cumulative PTF file.

Extract the Cumulative PTF Files

Each version of the Universal Agent for z/OS cumulative PTF distribution file is available in two formats:

- Compressed archive
- Compressed .zip file.

The contents of both files are the same. The different formats are provided for flexibility.

Compressed Archive

This file can be decompressed using any utility that can process files in Lempel-Ziv (that is, **compress**) format. The decompressed file's contents can be extracted using **tar**.

For example:

```
zcat sb-7.2.0.1-zos.PTFs.tar.Z | tar xvf -
```

This decompresses and extracts the 7.2.0.1 Universal Agent for z/OS distribution file that contains the cumulative PTF file on a UNIX-based system.

ZIP Archive

This file (for example, `sb-7.2.0.1-zos-PTFs.zip`) can be decompressed using any utility that can process files in **ZIP** (or, PKZIP) format.

To unzip the file, use a third-party ZIP-based extraction application or the native Windows file extraction tool available from Windows Explorer.

For example, to extract the files using Windows' File Explorer (or just Windows Explorer in some older Windows versions), right-click on the file name, select Extract All from the pop-up menu, and follow the instructions.

Transfer the Cumulative PTF Files

For the .XMT file(s), a binary transfer must be performed; otherwise, the XMIT data set(s) will not unpack correctly.

For the .JCL file, a text transfer must be performed.

Due to the numerous file transfer products available on the market, it is impractical to provide instructions for them all. FTP is one of the most popular, and the `UNVFTP.TXT` file included in the distribution file is a sample script that can be used as a reference to customize the transfer to your environment.

If you want to use `UNVFTP-PTF.TXT` (see the sample FTP script, below), begin by making the following changes:

Step 1	Change #HOSTNAME to the z/OS host name or IP address.
Step 2	Change #USERNAME to a z/OS user ID that has updated access to the XMIT data sets being updated.
Step 3	Change #PASSWORD to the z/OS user ID's password.
Step 4	Change #HLQ to the data set high level qualifier used to allocate the XMIT data sets.

```
open #HOSTNAME
user #USERNAME #PASSWORD

ascii
quote site blksize=3120 recfm=fb lrecl=80 pri=1 sec=1 cyl
put UNVRECV-PTF.JCL '#HLQ.UNV.V7R2M1.RECVPTF.JCL'

;Uncomment these lines when transferring from a
;Unix- or Windows-based system.
;put ascii/UNVRECV-PTF.JCL '#HLQ.UNV.V7R2M1.RECVPTF.JCL'

bin
quote site blksize=3120 recfm=fb lrecl=80 pri=1000 sec=100 cyl
del '#HLQ.UNV.V7R2M1.PTFS.XMIT'
put UNV72101.XMT '#HLQ.UNV.V7R2M1.PTFS.XMIT'

quit
```

After uploading, review the comments contained in `'#HLQ.UNV.V7R2M1.RECVPTF.JCL'` and make the necessary changes. Submit the job to receive and restore the install file.

Carefully review the FTP output to confirm that the file is transferred successfully.

Receive the Cumulative PTF Datasets

After transferring the cumulative PTF file to z/OS, you must unpack it using the JCL uploaded from `UNVRECV-PTF.JCL` (which should now reside in `#HLQ.UNV.V7R2M1.RECVPTF.JCL`).

The sample JCL to receive and restore the cumulative PTF file, below, shows the statements contained in `UNVRECV-PTF.JCL`. The job performs a TSO RECEIVE on the `UNV6511.XMT` file uploaded by `UNVFTP-PTF.TXT`.

The JCL includes a MODIFICATIONS section at the beginning of the file that describes what JCL modifications are required prior to submitting the job. Read and complete each of the listed modifications.

Submit the job on z/OS. Return code 0 is expected.

After the data set is successfully unpacked, it is longer required and can be deleted. The files then are ready to be installed.

```
//UNVRECV  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//             COND=(0,NE),REGION=(0M)
//*******************************************************************
//* (C) COPYRIGHT 2000-2021 STONEBRANCH, INC.  ALL RIGHTS RESERVED.
//*
//* STONEBRANCH, INC.
//* UNIVERSAL PRODUCTS
//*
//* UNVRECV
//*
//* DESCRIPTION
//* -----------
//* PERFORM A TSO RECEIVE ON THE TRANSMITTED DISTRIBUTION DATA SET
//* AND A DFHSMdss RESTORE TO PREPARE FOR INSTALLATION.
//*
//* ALL STEPS MUST END WITH RC 0.
//*
//* MODIFICATIONS
//* -------------
//* 1  MODIFY THE JOB STATEMENT TO MEET LOCAL REQUIREMENTS.
//*
//* 2  CHANGE ALL '#HLQ' TO THE HIGH-LEVEL QUALIFIER OF THE
//*    UNIVERSAL PRODUCTS DATA SETS.
//*
//* 3  CHANGE ALL '#VOLSER' TO THE VOLUME SERIAL NAME ON WHICH TO
//*    ALLOCATE THE DISTRIBUTION DATA SETS.
//*******************************************************************
//*
//DEL00    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE #HLQ.V721.UNV.PTFS
  SET MAXCC=0
/*
//*
//*******************************************************************
//* RECEIVE PTF DISTRIBUTION FILE
//*******************************************************************
//RECV     EXEC PGM=IKJEFT01
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTERM  DD  SYSOUT=*
//SYSTSIN  DD  *
  RECEIVE INDA('#HLQ.UNV.V7R2M1.PTFS.XMIT')
    DA('#HLQ.V721.UNV.PTFS') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    CYL SPACE(750 100) RELEASE
/*
//*
```

Installing a Cumulative PTF File

Cumulative PTF Maintenance is installed using members **UNVMTRCV**, **UNVMTAPL**, and **UNVMTACC** in the **INSTALL** product library.

The installation itself consists of running a number of batch jobs. The output of these batch jobs should be kept until a correct installation has been verified.

Step 1	Member **UNVMTRCV** performs an SMP/E RECEIVE of the PTFs. Change the data set allocated on ddname **SMPPTFIN** to the PTF data set. Modify the JCL to meet local requirements, and submit the job.
Step 2	Member **UNVMTAPL** performs an SMP/E APPLY of the PTFs. Modify the JCL to meet local requirements, and submit the job. The job should end with condition code 0. If the job does not end with condition code 0, it is most likely due to a HOLD on one of the PTFs. This is indicated by message GIM30206E printed on ddname **SMPOUT**. In this case, see PTF HOLDs, above.
Step 3	Fully test the maintenance. Only after all testing is completed should you proceed to the next step.

Step 4	Member **UNVMTACC** performs an SMP/E ACCEPT of the PTFs. Modify the JCL to meet local requirements, then submit the job. The job should end with condition code 0 or 4. If it does not, it is most likely due to a HOLD on one of the PTFs. This is indicated by message GIM30206E printed on ddname **SMPOUT**. In this case, see PTF HOLDs, above.

z/OS Installation - Sysplex

- Introduction
- Sysplex Solution
- UAG Sysplex
  - UAG Sysplex System View
  - CF List Type Structure
  - File Monitor Support for Secondary Agents
- Configuration Parameters Used for Sysplex Configuration
  - Parameters in UBRCFG00
  - Parameters in UAGCFG00
- z/OS Console Commands

Introduction

IBM provides the ability to cluster z/OS systems together using the IBM Sysplex technology, which is a combination of IBM hardware and software components. The individual z/OS systems are referred to as Sysplex members.

The IBM JES subsystem supports a Multi-Access Spool (MAS) configuration that allows for batch jobs to be distributed among participating JES subsystems. A JES MAS configuration may be used independently of a Sysplex environment or in combination with a Sysplex environment. When used in combination with a Sysplex environment, IBM recommends the JES MAS configuration match the Sysplex configuration.

The Universal Agent for z/OS Sysplex feature provides for the management of workload across all Sysplex members. This page describes the general architecture and design of the Universal Agent for z/OS Sysplex feature.

Sysplex Solution

From a workload management perspective, a z/OS Sysplex can be represented as a single z/OS image. A single-system view of the Sysplex is represented by a single Agent called the Primary agent which runs on any Sysplex member. Other agents, called Secondary agents, run on the other Sysplex members.

Note

The sysplex_role Universal Broker configuration option is used to select the Sysplex role for an agent.

Neither the Universal Controller nor the Universal Agent for z/OS participate in the distribution of workload across the sysplex images. The controller simply executes z/OS tasks on the Primary z/OS agent that represents the Sysplex.

A batch job submitted to JES on one z/OS system may be routed by JES or by IBM Workload Manager (WLM) to any one of the Sysplex z/OS members. The routing or distribution of batch workload is based on JCL specifications, system configuration and the state of the Sysplex members.

Universal Controller starts a z/OS task by sending a task start request to the Primary Universal Agent for z/OS. The Agent submits the requested job to JES. The job can potentially execute on any one of the Sysplex members. The Agents installed in the z/OS Sysplex cooperate with each other to manage the execution of the job.

Each Agent in the Sysplex can provide complete job management capabilities regardless of which Agent in the Sysplex submitted the job to JES.

Job management capabilities include:

- Automatic data set cleanup prior to job execution.
- Tracking the execution of the job and job steps.
- Collecting and retrieving the job's JES sysout data sets.

The z/OS Agents use the IBM Cross-System Coupling Facility (XCF) for Agent-to-Agent communication within the Sysplex. The Agents utilize the XCF data sharing capabilities for message passing and sharing of common data structures.

UAG Sysplex

UAG for z/OS will create and join a Sysplex group if it is running Sysplex aware.

The XCF group name will consist of the characters **UAG**, followed by the first 4 (upper-case) characters of the system ID (system_id from UBRCFG00).

Each UAG will have a member name, which will be the group name appended with @ and followed by the MVS system name.

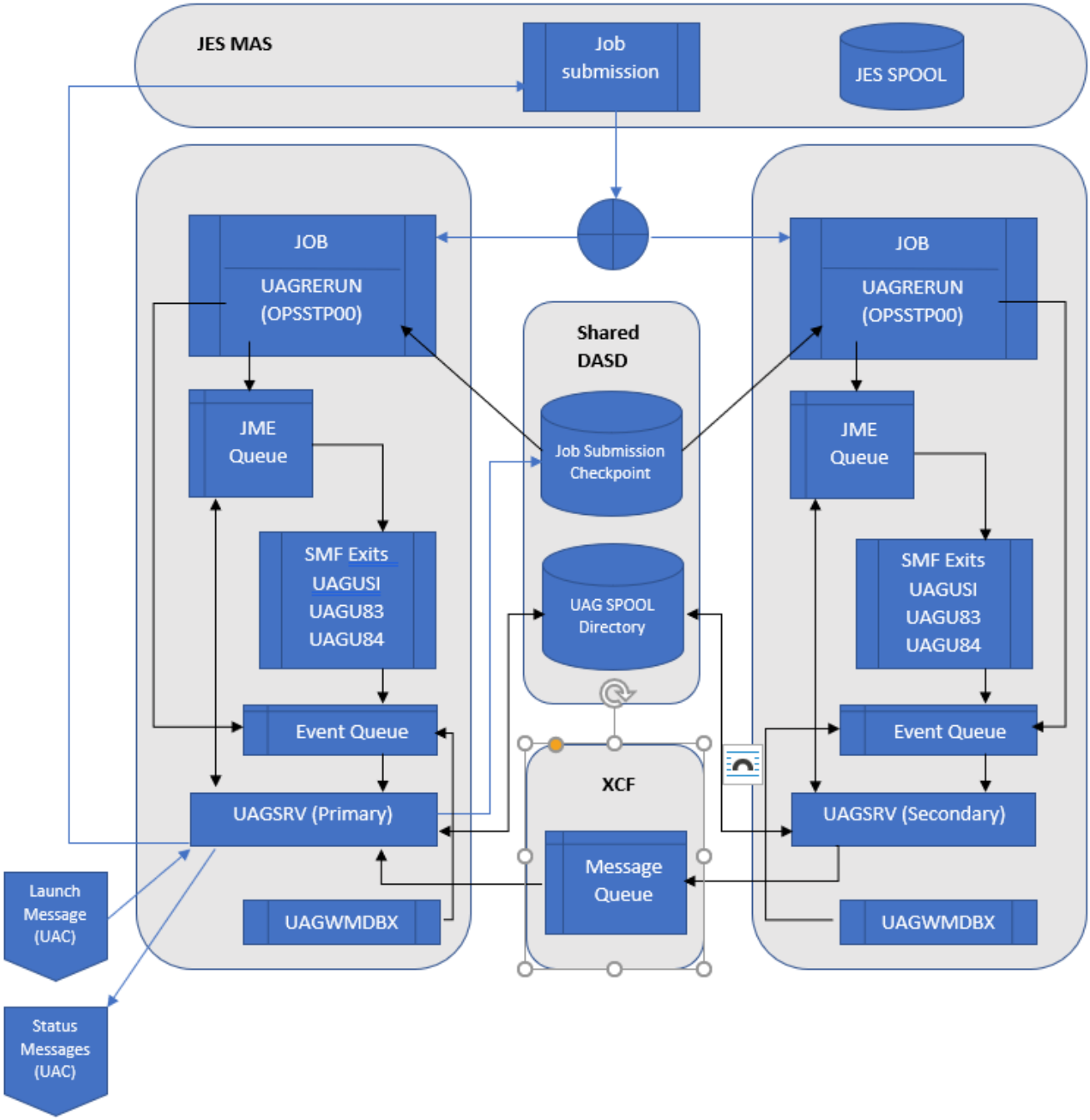
For example, UAG with a system ID of **mndv**, running on **DVZOS202**, would have a group name of **UAGMNDV** and a member name of **UAGMNDV@DVZOS202**.

UAG Sysplex System View

The Sysplex System View below illustrates the UAG deployment in a sample Sysplex environment. The Sysplex environment consists of two z/OS images, SYS1 and SYS2, and the Sysplex shared resources, JES, DASD, and XCF.

The following diagram illustrates a job submitted to one of two of the Sysplex members and the SMF exits that are called. The SMF exits reference the JME in ECSA and send events to the local UAGSRV via the event queue in z/OS High Common Storage.

1. A Launch message is received by the Primary UAG.
2. The Primary UAG writes a record to the Job Submission Checkpoint dataset, processes the JCL and submits it to the z/OS Internal Reader.
3. The JCL passes through JCL conversion and interpretation. The UAGUJV exit is invoked and sends an Event message to UAG to prompt it to look for JCL errors that might have prevented the job from entering the execution phase. (JCL conversion and interpretation can happen on different processors in the system depending on the system configuration.)
4. If a JCL error preventing the job from running is detected, a status message is sent to the Universal Controller and processing ends.
5. Once the job starts execution (on whatever system), program UAGRERUN gets control as the first step in the job. UAGRERUN performs pre-processing necessary to run and track the job on the local z/OS system. It creates the JME in ECSA to allow the SMF exits to track the job through Step Initiation, Step End and Job End processing.
6. As the job passes through Step End and Job End, Events are created on the Event Queue tracking the job's progress.
7. Events are removed from the Event Queue by the UAGSRV instance running on the system and processed.
8. In the case of a Primary UAG, Status messages are sent to the Universal Controller. In the case of a Secondary UAG, messages are queued to the XCF Message Queue. These messages are removed from the queue by the Primary UAG and Status messages are sent to the Universal Controller.
9. In both cases, any requested output is written to the UNVSPOOL directory for retrieval by the Universal Controller.
10. Any information required for eventual rerun processing is also sent to the Universal Controller.



Each of the components is described in the following table:

JES MAS	JES MAS (Multi Access Spool) environment (aka JESPLEX), provides for sharing JES resources between multiple z/OS images. The JES MAS environment can be implemented independent of Sysplex; however, IBM recommends that a JES MAS environment matches the Sysplex environment. A job is launched by UAG by submitting the job's JCL to JES. The JES subsystem manages the entire life of the job, from JCL conversion, interpretation, job execution, managing job output, to finally purging the job resources. In a Sysplex environment, when a job is submitted to JES, JES or WLM, the JES subsystem may decide to route the job to another Sysplex member for processing. The JES spool volumes, where jobs and job outputs are queued, is available to all members of the JES MAS.

Job Submission Checkpoint	The purpose of the JSC is to determine if a job should be tracked by UAG and to provide information to UAG to do so. JES may route the job to a different system for conversion/interpretation and execution. The system on which these steps take place cannot be determined prior to job submission. UAG utilizes shared DASD to maintain the Job Submission Checkpoint (JSC) data set. The dataset is a VSAM KSDS cluster. The Primary UAGSRV creates a job submission record in the JSC prior to submitting the job to JES. Note Use the JSC_DATASET UAG configuration option to specify the name of a VSAM Job Submission Checkpoint cluster. The VSAM cluster must be defined on a DASD volume that is available to all members in the Sysplex.
UAG SPOOL Directory	The UAG SPOOL directory is used by UAG to store spooled output for a job. At a minimum, the UAGRERUN report is stored here. Other datasets which might be stored here are the JESMSGLG, JESJCL and JESYSMSG datasets. UAG utilizes a shared zFS filesystem to mount the UNVSPOOL directory. This directory should be accessible to all members in the Sysplex. Note Use the SHARED_MOUNT_POINT and SHARED_MOUNT_POINT_MODE configuration options to specify the name and mode (access permissions) of the directory where the UNVSPOOL directory should be mounted. Note The JES_SYSOUT_RETENTION can be used to specify how long spooled output will be retained by UAG. Note The UNVSPOOL directory should be mounted as Sysplex aware by specifying the RWSHARE parameter on the MOUNT command. Failure to do so will result in a UNV3333E error during broker startup.
XCF Message Queue	XCF (Cross Coupling Facility) is a Sysplex component that provides services for communications and data sharing between Sysplex members. UAG utilizes a message communication channel using XCF services. Secondary UAGSRVs generate messages to the Primary UAGSRV to track the life cycle of a job. Secondary UAGSRVs do not communication directly with the Controller. The XCF Message Queue will survive across agent restart to preserve any tracking data. The XCF message Queue will be deleted if all agents in an agent group are shut down. Note Use the CF_STRUCT_NAME UAG configuration option to specify the name of a Coupling Facility structure that will be used to communicate from the Secondary Agents to the Primary Agent.
UAGRERUN	Program UAGRERUN is a component of UAG. A job step is inserted at the start of every job submitted by UAG which executes this program. UAGRERUN performs pre-processing necessary to run and track a job on a system. This includes, but is not limited to, creating the JME control block structure in ECSA to allow the UAG SMF exits to track the job. Note UAGRERUN needs to be available to every job submitted by UAG on every Sysplex member. This can be accomplished by adding the load library containing UAGRERUN to the z/OS linklist. Alternatively, the RERUN_LOAD_LIBRARY configuration option can be used to specify the name of the APF authorized load library which contains the UAGRERUN program. This library would need to be made available to all members of the Sysplex.
UAG SMF Exits	The UAG SMF exits are used to track a job submitted by UAG through its life cycle. • UAGUJV is used to detect JCL errors that prevent a job from executing. • UAGUSI is used to control job execution and to perform skip step processing. • UAGU83 is not used for job tracking. It is used for File Monitoring. • UAGU84 is used to track step end and job end events.
Event Queue	The Event Queue is an area in shared z/OS high common storage allocated for each agent. UAGRERUN and the SMF exits can queue event message to be consumed by the agent which owns the queue. The Event Queue is designed to survive across agent restarts to prevent loss of event data and to allow SMF exits to continue to provide tracking events in the event of an agent shutdown. The Event Queue will be deleted when z/OS is IPLed. Note The HIGH_COMMON_STORAGE configuration option can be used to limit to amount of high common storage used by an agent. When the limit is reached, further tracking events will be lost.
UAGWMDBX	UAGWMDBX is a z/OS WTO Message Data Block exit which looks for certain WTOs related to jobs submitted by UAG. For example: JCL errors.

CF List Type Structure

UAG uses a CF List type structure with the following values:

Setting	Value
List headers	1
Lock table entry count	1
Adjunct data	No
Alterable	Yes
Max number of list entries	20
Max number of data elements	160
Max number of data elements per entry	128
Reference option	None
Data element size descriptor	ElemIncrNum
Data element size value	2

Note

Users can alter only the `Max number of list entries` and `Max number of data elements` settings.

IBM provides a CFSIZER web tool (Structure type OEM List) which can be used to calculate the structure size. (Given the input above, this tool returned the INITSIZE and SIZE values of 9M.)

The structure name can be chosen by users and must be coded on the CF_STRUCT_NAME configuration option.

UAG uses this structure to communicate job tracking information from the Secondary agents to the Primary agent. List entries indicate events such as job start, step end and job end. List entries remain on the list until the primary agent has resources to process them. When the list structure is full, the secondary agents will wait until sufficient space is available before writing more tracking information. The required size of the structure is therefore dependent on the number of jobs being tracked, the number of job steps in those jobs and the resources available to the Primary agent to process the data.

File Monitor Support for Secondary Agents

File Monitors now function across a Sysplex. A File Monitor can be set, and the dataset Create, Change, or Delete will be detected on any system in the Sysplex where a UAG with the same system ID is running.

File Monitors will be detected while UAG is down as long as UAG was up when the File Monitor was set.

Exists and Missing File Monitors are resolved on the system where the Primary UAG is running. If a dataset is available only on a Secondary system, it will not be considered.

Configuration Parameters Used for Sysplex Configuration

Parameters in UBRCFG00

Name	Description
system_id	All Primary and Secondary Brokers that belong to the same group must have the same system_id.
sysplex_role	Select a value of primary for the primary agent; select a value of secondary for all others.
unix_spool_data_set	All Brokers that belong to the same group must reference the same dataset name. The dataset must reside on shared DASD that is available to all Sysplex members. This file system should not be shared among Brokers that are not part of the same Sysplex group.
mount_point	zFS mount point for non-shared UNIX file systems (currently only UNVDB). This mount point should not be shared between systems.

shared_mount_point	zFS mount point for shared UNIX file systems (currently only UNVSPOOL). All Brokers that belong to the same group must use the same directory. This mount point should be available to all members in the Sysplex. In non Sysplex situations, this parameter can be omitted, and it will default to the value specified for mount_point.
mount_point_mode	Mode (access permissions) to use during mount_point initialization.
shared_mount_point_mode	Mode (access permissions) to use during shared_mount_point initialization. In non-Sysplex situations, this parameter can be omitted; it will default to the value specified for mount_point_mode.

Parameters in UAGCFG00

Name	Description
jsc_dataset	All Primary and Secondary agents that belong to the same group must use the same UNVJSC VSAM cluster. This cluster must be allocated on shared DASD that is available to all members in the Sysplex. Agents that are not part of the Sysplex group must use a different VSAM cluster.
cf_struct_name	Name of the Coupling Facility structure which will be used to store the XCF Message Queue. All Primary and Secondary agents that belong to the same group must use the same structure. The structure should not be shared between agents that are not part of the same Sysplex group.

automatic_failover	If the value of the sysplex_role parameter in UBRCFG00 is primary or secondary, this parameter can be used to control automatic failover.
	
	Automatic failover allows a Secondary agent to become the Primary agent when the original Primary agent ends.
	
	Valid values:
	
	• **never**
	Automatic failover will not occur. Manual failover is still available.
	• **always_primary**
	For agents configured as Primary agents only. This agent should always be Primary. If another Primary agent is active when this agent starts, that agent will become a Secondary agent. If this agent cannot be a Primary agent, it will shut down.
	• **primary_secondary**
	For agents configured as Primary agents only. This agent will try to start as a Primary agent. If another Primary agent is already active, this agent will become a Secondary agent. It becomes first in the ranking to become Primary during failover.
	• **secondary_primary[n]**
	For agents configured as Secondary agents only: This agent will start as a Secondary agent. When the Primary agent ends, this agent is eligible to become the Primary agent.
	  • An optional integer [n] can be appended to this value. The integer controls the ranking of multiple Secondary agents during failover. A lower number means a higher priority in the failover ranking.
	  • When multiple agents have the same ranking, the agent that started earliest will be considered to have a higher ranking.
	  • Default for **[n]** is 1. The range is 1-32.
	
	Default is **never**.
	
	(Also see the AUTOMATIC_FAILOVER UAG configuration option.)

z/OS Console Commands

```
F <ubroker>,APPL=UAG,PRIMARY
```

This command causes an agent that is running in Sysplex Secondary mode to become a Primary agent until it is restarted or otherwise caused to become a Secondary agent.

If the agent is not running in Secondary mode, or a Primary agent is already active with the same system ID, the command will fail.

```
F <ubroker>,APPL=UAG,SECONDARY
```

This command causes an agent that is running in Sysplex Primary mode to become a Secondary agent until it is restarted or otherwise caused to become a Primary agent.

If the agent is not running in Primary mode, the command will fail.

```
F <ubroker>,APPL=SHUTDOWN, [ FAILOVER [ ,<sysname> ] | NOFAILOVER ]
```

When issued against a Secondary agent	This command behaves like the z/OS STOP command (P <ubroker>).
When issued against a Primary agent	This command shuts down the Broker (and agent) while controlling the Sysplex failover behaviour:
When issued without the FAILOVER or NOFAILOVER parameter	Failover will behave as configured by the automatic_failover parameter in UAGCFG00.
When FAILOVER Is specified	An available Secondary agent will take over as Primary, regardless of how failover is configured. When the optional <sysname> is specified, the agent running on the designated z/OS system will take over as Primary agent regardless of how fail over is configured.
When NOFAILOVER Is specified	No Secondary agent will take over as Primary, regardless of how failover is configured.

Note

Behaviour of the z/OS STOP console command with failover is identical to the F <ubroker>,APPL=SHUTDOWN command with no other parameters.

z/OS Installation - Time Zone Environment Variable

- Time Zone Environment Variable
- North American Values
- European Values
- Customizing the Start and End of Daylight Saving Time
- Additional Information

Time Zone Environment Variable

Universal Agent components execute in a z/OS UNIX environment, also known as a POSIX(ON) Language Environment.

The z/OS UNIX environment assumes that the z/OS system time is GMT or UTC format. It then uses the **TZ** environment variable value to determine the adjustments required for local time.

The **TZ** environment variable is an IBM Language Environment (LE) environment variable with a value set to the appropriate time zone and offset information so that time values are properly processed. **TZ** allows you to set the standard and daylight savings (or summer time) time zones and the offset from the local time zone to the UTC time.

LE environment variables, such as **TZ**, are set with the LE option **ENVAR** using the JCL step **PARM** keyword. For example, the following statement sets the standard time zone to Eastern Standard Time (EST) with an offset of 5 hours and the daylight saving time zone to Eastern Daylight Saving Time (EDT).

```
//PS1      EXEC PGM=UCMD,PARM='ENVAR(TZ=EST5EDT)/&UPARM'
```

Note

The forward slash (**/**) character separates the **LE** runtime parameters from the program parameters.

North American Values

TZ environment variables values for North America are listed below.

- EST5EDT: Eastern Standard Time, Eastern Daylight Time
- CST6CDT: Central Standard Time, Central Daylight Time
- MST7MDT: Mountain Standard Time, Mountain Daylight Time
- PST8PDT: Pacific Standard Time, Pacific Daylight Time
- AKST9AKDT: Alaska Standard Time, Alaska Daylight Time

European Values

TZ environment variables values for Europe are listed below.

- GMT0BST: Greenwich Mean Time, British Summer Time
- WET0WEST: Western Europe Time, Western Europe Summer Time
- CET-1CEST: Central Europe Time, Central Europe Summer Time
- EET-2EEST: Eastern Europe Time, Eastern Europe Summer Time

Other common time zone abbreviations may be used. IBM does not document all possible values.

Customizing the Start and End of Daylight Saving Time

An optional rule that specifies when daylight saving time begins and ends can be appended to the **TZ** environment variable value.

The format of this rule is `[,startdate[/time],enddate[/time]]`.

You must specify `startdate` and `enddate` in one of the following formats:

`Jn`	The Julian day, where n is in the range 1 to 365, inclusive. This format cannot handle leap days. J59 is always February 28 and J60 is always March 1.

`n`	The Julian day, where `n` is in the range 0 to 365, inclusive. This format does support leap days: • In non-leap years, day 59 is March 1 and day 364 is December 31. • In leap years, day 59 is February 29 and December 31 is day 365.
`Mm.n.d`	Where: • `m` is the month in the range 1 to 2, inclusive. • `n` is the week of the month in the range 1 to 5, inclusive. • `d` is the day of the week in the range 0 to 6, inclusive. Use week 5 to specify the last occurrence of a particular day within the specified month. Week 1 always refers to the first occurrence of a given day within the specified month. For example, M3.5.0 is the last Sunday in March and M11.1.0 is the first Sunday in November.

`time` has the format HH[:MM[:SS]]. If `time` is omitted from the rule, the default time is 02:00:00.

Additional Information

For more information on the **TZ** environment variable, see the z/OS Unix System Services Command Reference manual.

z/OS Installation - TCP/IP Configuration

- Overview
- Specifying TCP/IP Affinity
- TCP/IP Resource Protection
  - Stack Access Control
  - Port Access Control
  - Network Access Control
  - Socket Option Access Control

Overview

This page describes optional Universal Agent and TCP/IP configuration topics. Whether these steps are required or desirable depends on your local TCP /IP configuration.

Specifying TCP/IP Affinity

Universal Agent programs are considered generic client and server programs in IBM's TCP/IP terminology. They do not have an affinity for a specific transport provider (TCP/IP started task). The programs will utilize the appropriate transport provider based on TCP/IP configuration.

An affinity to a specific transport provider can be established for any Universal Agent program using Language Environment variable **_BPXK_SETIBMOPT_TRANSPORT**. The variable value specifies the TCP/IP started task name the program must use as its transport provider.

The JCL illustration below establishes affinity to TCP/IP started task TCPIPA:

```
//PS1    EXEC PGM=UCMD,
//       PARM='ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA")/&UPARM'
```

TCP/IP Resource Protection

The IBM TCP/IP product (Communications Server) offers optional protection to TCP/IP resources using SAF interfaces. The protection is implemented with a set of resource profiles defined in the SERVAUTH class.

If you are using the TCP/IP resource protection, you must permit appropriate privileges to the user profiles with which Universal Agent executes. Refer to the IBM Communications Server: IP Configuration Guide for complete details on TCP/IP resource protection. The TCP/IP resource profiles and the Universal Agent required access are discussed in the following sections.

Stack Access Control

The SAF resource profile **`EZB.STACKACCESS.sysname.tcpname`** in the SERVAUTH class controls which user profiles have access to a TCP/IP stack. All user profiles with which Universal Agent executes require READ access to the appropriate TCP/IP stack access profile.

Port Access Control

The SAF resource profile **`EZB.PORTACCESS.sysname.tcpname.SAF`** keyword in the SERVAUTH class controls access to specific non-ephemeral port ranges.

The Universal Broker binds to a service port (defaults to 7887). The user profile with which the Universal Broker started task executes requires READ access to any resource that protects this port.

The Universal Enterprise Controller binds to a service port (defaults to 8778). The user profile with which the Universal Enterprise Controller started task executes requires READ access to any resource that protects this port.

Network Access Control

The SAF resource profile **`EZB.NETACCESS.sysname.tcpname.zonename`** in the SERVAUTH class controls access to security zones. A security zone defines networks and hosts by IP address. All user profiles with which Universal Agent executes require READ access to the appropriate TCP/IP security zones profile.

Socket Option Access Control

The SAF resource profile **EZB.SOCKOPT.sysname.tcpname.SO_BROADCAST** in the SERVAUTH class controls access to the socket SO_BROADCAST option.

No Universal Agent programs use the SO_BROADCAST socket option, so no user profiles require access.

z/OS Installation - SAP RFC DLL

- Overview
- Background
- Output in the USS File System
- Universal Agent Components

Overview

The SAP RFC DLL is a remote function call library provided by SAP AG for applications to interface with the SAP system. A number of Universal Agent components make use of the SAP RFC DLL to provide SAP-related functionality.

The SAP RFC DLL interacts with the z/OS Language Environment (LE) and the z/OS UNIX System Services (USS), resulting in residual files being created by LE in the USS file system.

Background

A z/OS user profile must have a properly defined OMVS segment in order to run a program that utilizes USS services. The OMVS segment specifies, among other attributes, a home directory in the USS file system. If no home directory is specified, it defaults to **/tmp**.

The z/OS Language Environment (LE) produces output under certain conditions. When a LE batch job or started task is executed, the LE MSGFILE option specifies the ddname to which LE output is written. By default, MSGFILE specifies the SYSOUT ddname.

Additionally, for diagnostic purposes, LE can write LE dumps and traces. As a batch job or started task, LE writes the dumps and traces to ddname CEEDUMP. When a LE program is run from the USS environment, the LE MSGFILE option defaults to standard error, and LE dumps and traces are written to a file created in the current working directory with a name starting with CEEDUMP.

The SAP RFC DLL is loaded dynamically at run time when a Universal Agent batch job or started task needs to use SAP RFC functions. When the DLL is loaded, RFC initialization is performed by the DLL. As part of the RFC initialization, a USS **popen** function is called from within the DLL that creates two USS processes that run as child processes of the Universal Agent program. The child processes run for a very brief amount of time (1-2 seconds).

A USS child process inherits a number of attributes from its parent, which in this case is a Universal Agent program. Among the attributes inherited are the user ID profile, including the OMVS segment, as well many of the LE options specified in the batch job or started task. Among the attributes that a child process does not inherit is the ddname allocations of its parent. The child processes run in a separate address space managed by z/OS Workload Manager.

Output in the USS File System

The two child processes created by the **popen** function executed by the SAP DLL can result in LE creating certain files in the USS file system. The LE options that the batch job or started task execute with are inherited by the child processes created by the **popen** function. When the LE options produce output, this output will be created for each of the USS child processes in the USS file system.

As an example, the LE option RPTOPTS(ON) will write a report to the location specified by the LE MSGFILE option. The report lists all the LE option values and the source of the options. The two child processes created by the SAP RFC DLL inherit the LE options, and when both of the child processes end, LE writes a options report to the location specified by the LE MSGFILE option.

Since the child processes run in a USS environment, the LE MSGFILE option defaults to standard error; however, there is no standard error defined in this case, so LE writes its output to a file named SYSOUT in the current working directory. The current working directory will be the home directory specified in the user profile OMVS segment.

There are a number of conditions that cause LE to produce output. A partial list is:

- Certain LE options produce reports to the MSGFILE location. For example, RPTOPTS and RPTSTG.
- LE error messages resulting from invalid options or run-time errors are written to the MSGFILE location.
- LE dumps and traces produced due to run-time exceptions or the LE TRACE option. Dumps and traces are written to either the CEEDUMP ddname or to a USS file name starting with CEEDUMP.

Universal Agent Components

The following Universal Agent components use the SAP RFC DLL:

- Universal Broker
- Universal Enterprise Controller
- Universal Connector

When these components are executed with LE options that product output, a USS file named SYSOUT or a file starting with CEEDUMP will be created in the home directory of the user profile executing the component.

If two or more instances of a component executing with different user profiles share the same home directory, security violations may occur. The first component creates a USS file named SYSOUT owned by its user ID and the second component executing with a different user profile attempts to append to it resulting in a security violation.

To avoid the potential security violation when attempting to append to the LE SYSOUT file on the USS file system, define each z/OS user profile with a unique home directory.

z/OS Installation - Configuration of z/OS System SSL

- Configuration of z/OS System SSL
  - SSL Benefits
  - Required Conditions for Using SSL
- Additional Information

Configuration of z/OS System SSL

Universal Agent can use the IBM z/OS System SSL library or the OpenSSL SSL library for its SSL/TLS network communications. The SSL library selection is made with the Universal Agent SSL_IMPLEMENTATION configuration option.

z/OS System SSL requires the IBM System SSL Cryptographic Services base element. In addition, Universal Agent requires Cryptographic Services Security Level 3 element, which includes the cryptographically strong SSL/TLS cipher suites.

SSL Benefits

System SSL provides the following benefits:

- Utilizes any cryptographic hardware features available reducing the amount CPU resources used by Universal Agent.
- Seamless integration with RACF certificate management features.

Note

If RACF digital certificates are new to you or your site, refer to the following documentation for complete details:

- z/OS Security Server RACF Security Administrator's Guide
- z/OS Security Server RACF Command Language Reference

Required Conditions for Using SSL

In order for Universal Agent to use z/OS System SSL, the following conditions must be met:

1. Universal Agent supports z/OS System SSL on z/OS 1.4 and above.
2. Universal Agent component SSL_IMPLEMENTATION configuration values must be set to **system**.
3. User profiles with which the Universal Agent component executes must have READ access to the RACF profile **IRR.DIGTCERT.LISTRING** in the FACILITY class.
4. User profiles with which the Universal Agent component executes must have a certificate key ring associated with them that includes the user's certificate and the CA's certificate.

Additional Information

The following pages provide additional detailed information for Configuration of z/OS System SSL:

- Integrated Cryptographic Service Facility (ICSF)
- Universal Broker Digital Certificate (RACF) Set-up

Integrated Cryptographic Service Facility (ICSF)

Integrated Cryptographic Service Facility (ICSF)

z/OS System SSL will use ICSF when available. The ICSF started task must be running and ICSF configuration completed.

The user profile with which the System SSL application executes must have access to the following ICSF resources in the CSFSERV class:

- CSFCKI, clear key import
- CSFCKM, multiple clear key import
- CSFDEC, symmetric key decrypt
- CSFDSG, digital signature generate
- CSFDSV, digital signature verify
- CSFENC, symmetric key encrypt
- CSFPKB, PKA key build
- CSFPKD, PKA decrypt
- CSFPKE, PKE encrypt
- CSFPKI, PKA key import

In addition to the resources above, it also may be necessary to grant the user access to the CSFIQA resource. That resource permits queries on the cryptographic hardware available on a particular system. If the user does not have READ access to that resource, System SSL processing can continue, but may do so with reduced functionality (for example, TLS 1.2 ciphers may not be accessible). ICH408I messages will also appear in the system log for each query that fails.

To use ICSF's random number generator - used to seed key generation – the CSFRNG resource must be defined to the CSFSERV class and the user must have READ access to it. If this resource is not available, or the user cannot access it, a less secure generator may be used.

Refer to the IBM z/OS ICSF Administrator's Guide for managing access to ICSF resources.

Universal Broker Digital Certificate (RACF) Set-up

Universal Broker Digital Certificate (RACF) Set-up

Setting up a digital certificate infrastructure in a production environment for the first time requires careful planning by the Security organization and Universal Agent administrator. The instructions provided on this page have been simplified for illustration purposes.

You work with RACF Digital Certificates using the RACF command **RACDCERT**. RACF profiles control access to the functions provided by **RACDCERT**.

The user profile with which the following commands are executed require either:

- SPECIAL attribute
- Appropriate access to the **IRR.DIGTCERT.function** profile in the FACILITY class.
  - READ access is required to **IRR.DIGTCERT.function** to issue **RACDCERT** commands for the executing user.
  - UPDATE access is required to **IRR.DIGTCERT.function** to issue **RACDCERT** commands for other users.
  - CONTROL access is required to **IRR.DIGTCERT.function** to issue **RACDCERT** command for SITE or CERTAUTH certificates.

The member **UNVINDC** in the **INSTALL** library contains the JCL to execute the RACF commands listed in the following steps.

Step 1	Create a Certificate Authority (CA) certificate and private key using the following RACDCERT command:
	
	`RACDCERT CERTAUTH GENCERT +`
	`  SUBJECTSDN(CN('Certificate Authority') +`
	`  OU('Security') +`
	`  O('Company Name, Inc.') +`
	`  C('US')) +`
	`  NOTAFTER(DATE(2030-01-01)) +`
	`  *KEYUSAGE(HANDSHAKE CERTSIGN) +`
	`  WITHLABEL('Company CA')`
	
	Change the subject and label names to meet local requirements.

Step 2	Create a certificate for the Universal Broker STC and sign it with the CA certificate created in Step 1 using the following RACDCERT command:
	
	`RACDCERT ID(UBRUSR) GENCERT +`
	`  SUBJECTSDN(CN('broker.company.com') +`
	`  OU('Operations') +`
	`  O('Company Name, Inc.') +`
	`  C('US')) +`
	`  KEYUSAGE(HANDSHAKE) +`
	`  WITHLABEL('Broker') +`
	`  SIGNWITH(CERTAUTH LABEL('Company CA')`
	
	Change the subject and label names to meet local requirements. The subject's Common Name (CN) value should uniquely identify this instance of the broker in the enterprise.

Step 3	Create a certificate key ring for the user profile UBRUSR with the following RACDCERT command: `RACDCERT ID(UBRUSR) ADDRING(BROKER)`

Step 4

Connect the CA certificate and the Universal Broker certificate to the key ring with the following RACDCERT command:

```
RACDCERT ID(UBRUSR) CONNECT(CERTAUTH LABEL('Company CA') +
   RING(BROKER)
RACDCERT ID(UBRUSR) CONNECT(LABEL('Broker') RING(BROKER) DEFAULT)
```

Change the labels to match the values used in previous steps.

Step 5

If the resource profile **IRR.DIGTCERT.LISTRING** in the FACILITY class is not defined, define it with the following RDEFINE command:

```
RDEFINE FACILITY (IRR.DIGTCERT.LISTRING) UACC(NONE)
```

Step 6

Permit the Broker user profile UBRUSR READ access to the RACF profile **IRR.DIGTCERT.LISTRING** in the FACILITY class using the following PERMIT command:

```
PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
```

Step 7

Modify the Universal Broker configuration member **UBRCFG00** as follows:

```
ssl_implementation   system
saf_key_ring        BROKER
```

Step 8

The CA certificate must be distributed to the remote systems from which Universal Agent managers are executed. The managers must be configured with the CA certificate in their list of Trusted CA certificates using the CA_CERTIFICATES configuration option.

The CA certificate is exported out of the RACF data base into a data set in a PEM (or base64) format with the following RACDCERT command:

```
RACDCERT CERTAUTH EXPORT (LABEL('Company CA')) +
   DSN(TEST.CA.CERT) FORMAT(CERTB64)
```

Change the label to match the value used in previous steps.

The **tsoprefix.TEST.CA.CERT** data set contains a PEM formatted certificate. The format is a text format that transfers safely across the network in text mode.

Note that the CA private key is not exported. The CA certificate does not contain any private data.

z/OS Installation - Configuration of Security

Introduction

Some Universal Agent components utilize z/OS security services to control access to product functions and resources. These pages describe the installation steps to activate product security and define resource controls.

Note

The use of the security resource is optional. If you do not want to use it at this time, this information can be skipped.

Detailed Information

The following pages specify the configuration methods for IBM's RACF and Computer Associate's ACF2 security products:

- RACF Class
- ACF2 Class
- Universal Command Security
- Universal Control Security
- Universal Event Monitor Security

Configuration of Security - RACF Class

RACF Class

All components use the Universal Agent RACF class **$UNV**. The class name can be changed if local requirements require it.

To install the Universal Agent RACF class, perform the following steps:

Step 1	Modify your local RACF class descriptor table by adding the following entry to **ICHRRCDE**:
	
	`ICHERCDE CLASS=$UNV,  +`
	`CASE=ASIS,  +`
	`FIRST=ANY,  +`
	`OTHER=ANY,  +`
	`MAXLNTH=246,  +`
	`DFTUACC=NONE,  +`
	`ID=128,  +`
	`POSIT=128`

Step 2	Modify the ID and POSIT values to meet local requirements. Support for the CASE parameter was added in z/OS 1.2. Case-sensitive profile names are necessary for USS path name support in the standard I/O access profiles. See Universal Command Standard I/O Access Profile for details. Care must be taken when defining case-sensitive profile names so that the profile characters are typed with the desired case. RACF will not automatically upper case the profile names. If the wrong case is used, the desired profile will not be found. If the standard I/O profiles are not necessary or USS path name support is not necessary, the CASE parameter can be removed. Sample assemble and link edit JCL is provided in member **UNVRRCDE** in the **INSTALL** library. IBM provides a sample ICHRRCDE table and the JCL to assemble and link edit it in **SYS1.PARMLIB(RACTABLE)**.

Step 3	Modify your local RACF routing table by adding the following entry to ICHRFR01.
	
	`ICHRFRTB CLASS=$UNV,  +`
	`ACTION=RACF`
	
	Sample assemble and link edit JCL is provided in member **UNVRFR01** in the **INSTALL** library.
	
	IBM provides a sample ICHRRCDE table and the JCL to assemble and link edit it in **`SYS1.PARMLIB(RACTABLE)`**.

Step 4	If you changed the default RACF class name **$UNV** to another name, the UCMD, UCTL, and UEM load modules must be modified to include the same class name. (If you did not change the default class name, skip this step.) 1. Member **UCMCMVS** in the **INSTALL** library contains an SMP/E USERMOD to apply the modification to the UCMD load module. Change the USERMOD to the new class name and submit the job. 2. Member **UCTCMVS** in the **INSTALL** library contains an SMP/E USERMOD to apply the modification to the UCTL load module. Change the USERMOD to the new class name and submit the job. 3. Member **UEMCMVS** in the **INSTALL** library contains an SMP/E USERMOD to apply the modification to the UEM load module. Change the USERMOD to the new class name and submit the job. All steps must end with return code 0.
Step 5	IPL the z/OS system to install the new class descriptor and routing tables.

Step 6	Activate the new class and define it for generic profile checking by entering the following RACF SETROPTS commands:
	
	`SETROPTS CLASSACT($UNV)`
	`SETROPTS GENERIC($UNV)`

Configuration of Security - ACF2 Class

ACF2 CLASS

All components use the Universal Agent ACF2 class **$UNV**. The class name may be changed if local requirements require it.

To install the Universal Agent ACF2 class, perform the following steps:

Step 1	Create a CLASMAP with the following ACF2 command:
	
	`SET CONTROL(GSO)`
	`INSERT CLASSMAP.$UNV RESOURCE($UNV) RSRCTYPE(UNV) ENTITYLN(246)`

Step 2	Rebuild and refresh the directories with the following console command: `F ACF2,REFRESH(CLASMAP),SYSID(*sysid*),CLASS(C),TYPE(GSO)`

Step 3	Define the following resource rules for Universal Command and Universal Control with the following commands:
	
	`SET RESOURCE(OSM)`
	`COMPILE * STORE`
	`  $KEY(UCMD) TYPE(UNV)`
	`  UID(ucmd_admin_id) ALLOW`
	`  - UID(ucmd_admin_id) SERVICE(READ) ALLOW`
	
	`SET RESOURCE(OSM)`
	`COMPILE * STORE`
	`  $KEY(UCTL) TYPE(UNV)`
	`  UID(uctl_admin_id) ALLOW`
	`  - UID(uctl_admin_id) SERVICE(READ) ALLOW`

Configuration of Security - Universal Command Security

- Overview
- Universal Command Remote Access Profile
- Universal Command Standard I/O Access Profile
- Universal Command Security Profile Definitions
  - Example 1
  - Example 2
  - Example 3
  - Example 4

Overview

Universal Command access controls consist of the following profile types:

- Remote Access profiles control Universal Command's access to remote systems.
- Stdio Access profiles control Universal Command's access to local data sets used as standard input, output or error.

Universal Command Remote Access Profile

The remote access profile controls Universal Command Manager's access to specific remote systems running Universal Agent. The remote system is identified by the IP address of the remote system, the port number on which the Manager is connecting to the remote system, and the remote system user ID with which the Manager is requesting the command to be executed.

Universal Command Manager identifies a remote system with the REMOTE_HOST and REMOTE_PORT configuration options, and the remote user ID with the USER_ID configuration option.

The remote access profile has the following format:

```
UCMD.Iipaddress.Pport.userid
```

The following table identifies the fields that comprise the profile name.

Field	Description
ipaddress	Numeric dotted-form IP address of the remote host as specified by the REMOTE_HOST option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address **256.10.2.123** is formatted as **256010002123**.
port	Numeric port number on which Universal Broker is listening as specified by the REMOTE_PORT option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port **7887** is formatted as **07887**.
userid	Remote user identifier with which Universal Command Manager will execute the remote command as specified by the USER_ID option. Whether or not a user identifier is required depends on the Universal Command Server configuration. If no user identifier is specified for the Manager, the userid value is <BLANK>. The value userid is upper case. Some remote hosts may have case-sensitive user identifiers. All user identifiers specified on the Manager are translated to upper case for building the profile.

Universal Command Standard I/O Access Profile

The standard I/O (**stdio**) access profile controls Universal Command Manager's access to data sets allocated for standard I/O. The standard I/O ddnames are **UNVIN** for standard input, **UNVOUT** for standard output, and **UNVERR** for standard error, unless otherwise changed with the SIO_LOCAL_FILE configuration option.

The standard I/O access profile has the following format:

```
UCMD.stdio.Iipaddress.Pport.allocation
```

The following table identifies the fields that comprise the profile name.

Field	Description

stdio	Standard I/O file which the profile is for. Valid values are: • STDIN for the standard input file. • STDOUT for the standard output file. • STDERR for the standard error file.
ipaddress	Numeric dotted-form IP address of the remote host as specified by the REMOTE_HOST option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address **256.10.2.123** is formatted as **256010002123**.
port	Numeric port number on which Universal Broker is listening as specified by the REMOTE_PORT option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port **7887** is formatted as **07887**.

allocation	Entity that is allocated to the standard I/O ddname. There are a number of different items that may be allocated to a ddname. The following formats are available:
	• A data set allocation is represented by **Ddsn**, where **dsn** is the data set name. For example, a ddname allocation of **PROD.APPL.DATA** is formatted as **DPROD.APPL.DATA**.
	  • If allocating a PDS and a member name is specified, do not include the member name in the profile name, only the PDS name.
	  • If allocating a relative generation data set, do not include the relative number in the profile name, only the GDG name.
	• A JES SYSIN, SYSOUT or SUBSYS= allocation is represented by the value SUBSYS.
	• A temporary data set allocation is represented by the value TEMPORARY.
	• A DUMMY or NULLFILE allocation is represented by the value NULLFILE.
	• A USS path name allocation is represented by **Upath**, where **path** is the USS path name. For example, a ddname allocation of **/prod/appl/data** is formatted as **U/prod/appl/data**.
	  • USS path name support starts at z/OS 1.4. Prior to z/OS 1.4, the path name is not available to format the profile name. The value used for path in this case is **UNKNOWNUSSPATHNAME**.
	  • USS path names are case sensitive. The Universal Agent class must be defined with the CASE=ASIS parameter to support case sensitive profile names.
	
	  The maximum length of the profile name is 246 characters. The maximum length of a USS path name allocated to a ddname is 256 characters. It is possible the path name may be truncated in formatting the profile name. Truncation does not result in an error condition.

Universal Command Security Profile Definitions

These examples illustrate sample RACF commands that can be used to define Universal Command security profiles and permit z/OS user identifiers access to those profiles. Refer to the IBM RACF documentation for complete details on RACF commands.

Example 1

Assume that you want to restrict Universal Command Manager for z/OS access to remote host **10.23.90.2**. The following profile would restrict access to only those z/OS users who have read access to the profile **UCMD.I010023090002.*.***.

The following TSO commands define the required profile and permits access to TSO user TSO555.

```
RDEF $UNV (UCMD.I010023090002.*.*) UACC(NONE)
PE UCMD.I010023090002.*.* CLASS($UNV) ID(TSO555) ACCESS(READ)
```

Example 2

Assume that you run all Universal Brokers on privileged port **1000**. To enforce the policy that z/OS Managers connect only to port 1000, define profile **UCMD.*.*.*** with universal access none and define **UCMD.*.P01000.*** with universal access read.

The following TSO commands define the required profiles.

```
RDEF $UNV (UCMD.*.*.*) UACC(NONE)
RDEF $UNV (UCMD.*.P01000.*) UACC(READ)
```

Example 3

Assume that you want to restrict root access to all hosts from Universal Command Manager for z/OS. The following profile would restrict root access to only those z/OS users who have read access to the profile **UCMD.*.*.ROOT**.

The following TSO command defines the required RACF profile.

```
RDEF $UNV (UCMD.*.*.ROOT) UACC(NONE)
```

Example 4

Assume that you want to restrict Universal Command Manager access to data sets **PROD.***. You also don't want Universal Command Manager to use any temporary data sets. The following profiles would restrict access to only those z/OS users who have read access to the profile.

The following TSO command defines the required RACF profile.

```
RDEF $UNV (UCMD.STD*.*.*.DPROD.**) UACC(NONE)
RDEF $UNV (UCMD.STD*.*.*.TEMPORARY) UACC(NONE)
```

Configuration of Security - Universal Control Security

- Overview
- Universal Control Remote Access Profile Format
- Universal Control Security Profile Definition
  - Example 1

Overview

Universal Control access controls consist of the following profile types:

Remote Access profiles control Universal Control's access to remote systems.

Universal Control Remote Access Profile Format

The remote access profile controls Universal Controls Manager's access to specific remote systems running Universal Agent. The remote system is identified by the IP address of the remote system, the port number on which the Manager is connecting to the remote system, and the command to be executed.

Universal Control Manager identifies a remote system with the REMOTE_HOST and REMOTE_PORT configuration options, and the command as one of the command options.

The remote access profile has the following format:

`UCTL.Iipaddress.Pport.command`

The profile name is composed of the following fields.

Field	Description
ipaddress	Numeric dotted-form IP address of the remote host as identified by the REMOTE_HOST option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address **256.10.2.123** is formatted as **256010002123**.
port	Numeric port number on which Universal Broker is listening as identified by the REMOTE_PORT option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port **7887** is formatted as **07887**.
command	Universal Control command that the Manager is requesting execution. Possible command values are START, STOP, and REFRESH.

Universal Control Security Profile Definition

This example illustrates sample RACF commands that can be used to define Universal Control security profiles and permit z/OS user identifiers access to those profiles. Refer to the IBM RACF documentation for complete details on RACF commands.

Example 1

Assume you wish to restrict Universal Control Manager for z/OS access to remote host **10.23.90.2**. The following profile would restrict access to only those z/OS users who have read access to the profile **`UCTL.I010023090002.*.*`**.

The following TSO commands define the required profile and permits access to TSO user TSO555.

```
RDEF $UNV (UCTL.I010023090002.*.*) UACC(NONE)
PE UCTL.I010023090002.*.* CLASS($UNV) ID(TSO555) ACCESS(READ)
```

Configuration of Security - Universal Event Monitor Security

- Overview
- Universal Event Monitor Remote Access Profile Format
- Universal Event Monitor Security Profile Definition
  - Example 1

Overview

Universal Event Monitor access controls consist of the following profile types:

Remote Access profiles control Universal Event Monitor's access to remote systems.

Universal Event Monitor Remote Access Profile Format

The remote access profile controls Universal Event Monitor Manager's access to specific remote systems running Universal Agent. The remote system is identified by three elements:

1. System's IP address
2. Port number that the Manager uses to connects to a Universal Broker executing on the system
3. User account specified from the UEM Manager, which is defined on the remote system

The Universal Event Monitor Manager identifies a remote system with the REMOTE_HOST and REMOTE_PORT configuration options, and the remote user account with the USER_ID configuration option.

The remote access profile has the following format:

`UEM.Iipaddress.Pport.userid`

The profile name is composed of the following fields.

Field	Description
ipaddress	IP address of the remote host, in dotted-decimal notation, as identified by the REMOTE_HOST configuration option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address **256.10.2.123** is formatted as **256010002123**.
port	Numeric port number on which Universal Broker is listening as identified by the REMOTE_PORT configuration option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port **7887** is formatted as **07887**.
userid	ID of a remote user account, which was specified by the Manager with the USER_ID option. If the Universal Event Monitor Server is configured to not require a user ID, the value for this field is <BLANK>.The value *userid* is upper case. Some remote hosts may have case-sensitive user identifiers. All user identifiers specified on the Manager are translated to upper case for building the profile.

Universal Event Monitor Security Profile Definition

This example illustrates sample RACF commands that can be used to define Universal Event Monitor security profiles and permit local user accounts access to those profiles. Refer to the IBM RACF documentation for complete details on RACF commands.

Example 1

Assume you wish to restrict Universal Event Monitor Manager for z/OS access to remote host **10.23.90.2**. The following profile would restrict access to only those z/OS users who have read access to the profile **`UEM.I010023090002.*.*`**.

The following TSO commands define the required profile and permits access to TSO user TSO555.

```
RDEF $UNV (UEM.I010023090002.*.*) UACC(NONE)PE UEM.I010023090002.*.*
CLASS($UNV) ID(TSO555) ACCESS(READ)
```

z/OS Installation - Performance Guidelines

- Overview
- UNIX System Services and Language Environment
- Universal Agent Managers
- Universal Broker and Universal Agent Servers
- Universal Enterprise Controller

Overview

Universal Agent consists of product components distributed throughout the enterprise communicating with each other over the computer network using the TCP/IP communication protocol.

Universal Agent offers reliable, fault tolerant, secure, and efficient communications between its distributed components. In order for product components to effectively meet their communication requirements, z/OS must provide sufficient execution time for the product components.

The execution of the communication protocol is a real-time activity and communication time-outs may be exceeded if product components are not dispatched appropriately while executing the communication protocol.

The following sections provide performance guidelines for Universal Agent for z/OS components.

UNIX System Services and Language Environment

All Universal Agent components are written in C/C++ and utilize z/OS Language Environment (LE) and z/OS UNIX System Services (USS).

The IBM z/OS UNIX System Services Planning manual includes a "Tuning Performance" chapter that should be reviewed to improve USS performance in general.

Universal Agent components do not attempt to use the USS **setpriority**, **chpriority**, or **nice** functions to adjust their performance group or service class.

Universal Agent Managers

Universal Agent Managers consist of Universal Command, Universal Data Mover, and Universal Event Monitor managers. They typically execute in the JES subsystem as a batch job or the OMVS subsystem as a USS shell command.

The managers communicate with remote Universal Brokers and their corresponding Universal Agent Server components on remote systems using the TCP /IP protocol.

In cases where the z/OS workload requires more resources than are available, z/OS will favor workload with a higher dispatch priority over workload with a lower dispatch priority. If a Universal Agent manager is being executed with a lower dispatch priority than other workload competing for resources, it may not be given sufficient execution time to meet its network communication timing requirements. The result will be false network time-out errors in the Universal Agent manager.

The effect of a network time-out condition depends on whether or not the Universal Agent manager is using the Network Fault Tolerant (NFT) feature. If NFT is used, the manager reestablishes the communication session and continues; otherwise, the manager ends with an error.

False network communication time-out errors can be addressed using one or both of the following options:

1. Increase the manager's NETWORK_DELAY configuration option value (default is 120 seconds). NETWORK_DELAY specifies the maximum amount of time to wait for data on a communication session before considering the session timed out. Increasing the value allows for the manager batch job to be swapped out for a longer period of time before the session will be considered timed out.
   However, in cases where a condition truly exists in the network that would result in a true network time-out, a larger NETWORK_DELAY value would result in a longer period of time before the manager would detect and respond to the time-out condition.
2. Raise the Universal Agent manager workload dispatch priority by placing it in a higher performance group or service class. Raising the workload dispatch priority will allow z/OS to provide sufficient CPU resources to the manager to meet network timing requirements.

Universal Broker and Universal Agent Servers

The Universal Broker started task is the center of activity on each system on which Universal Agent is installed. Almost all components communicate with a locally installed Universal Broker during their execution, including managers and servers.

Universal Broker is responsible for managing Universal Agent servers that are performing work on z/OS on behalf of remote Universal Agent managers. Universal Agent servers are created by the Universal Broker using the USS spawn function. The servers run as USS child processes of the Universal Broker started task in the OMVS subsystem.

The Universal Broker started task must execute with a sufficiently high performance group or service class in order to service all manager and server requests in a timely manner to avoid false network time-out conditions.

On heavily loaded systems, it is recommended to make the Universal Broker started task non-swappable to help overall improvement of Universal Broker.

Universal Enterprise Controller

The Universal Enterprise Controller (UEC) started task performs real-time monitoring of Agents distributed throughout the network. UEC communicates with each Agent on a defined polling interval.

UEC is a USS, multi-threaded application written in C/C++ that heavily utilizes TCP / IP. The amount of work that UEC performs depends directly on the number of Agents defined to it. UEC maintains Agent status information and Universal Event Subsystem information in the UEC database. The database is an HFS or zFS database that is mounted in the z/OS UNIX file system.

UEC must execute with a sufficiently high performance group or service class in order to perform its Agent monitoring service effectively. False Agent time-out alerts can result if UEC is not dispatched in a timely manner.

On heavily loaded systems, it is recommended to make the UEC started task non-swappable to help overall performance of UEC.

z/OS Installation - Converting STC User Profiles to a Non-Zero UID

- Overview
- Converting Universal Broker User Profile to Non-Zero UID
- Running Without BPX.SUPERUSER and BPX.DAEMON Access
- Converting Universal Enterprise Controller (UEC) User Profile to Non-Zero UID

Overview

Prior to Stonebranch Solutions 4.2.0, the Universal Broker and Universal Enterprise Controller (UEC) started task user profiles were required to have an OMVS UID value of 0. As of 4.2.0, the products were enhanced to execute with a user profile defined with a non-zero UID value to improve upon the product security features.

A Universal Agent installation that already has a user profile with UID 0 in use can convert the user profile from UID 0 to a non-zero UID value. There are a number of concerns when changing a user profiles UID value. The UID value identifies the user profile in the z/OS UNIX (USS) environment.

The following sections describe how to convert a Universal Broker or Universal Enterprise Controller user profile UID value from 0 to non-zero.

Converting Universal Broker User Profile to Non-Zero UID

The conversion steps assume the following:

- The UID value is being changed from 0 to 5001. If a UID value of 5001 does not work in your local environment, change all references to 5001 in the following steps to a unique, non-zero UID value suitable for your local environment. Note that the UID value must be unique among all user profiles.
- The Universal Broker user profile name is **UBRUSR**. If the Universal Broker STC in your local environment uses a different user profile name, change all references to **UBRUSR** in the following steps to the user profile name used in your local environment.
- The user ID used to execute the commands requires an OMVS segment.
- For full feature support, the user ID must have either UID 0 or READ access to the BPX.SUPERUSER profile in the FACILITY class. Starting with Universal Agent 6.5.0.0, a majority of agent features can be accessed without READ access to the BPX.SUPERUSER and BPX.DAEMON profiles. Refer to information below and throughout this section for more information on executing without access to those profiles.
- The Universal Broker HFS or zFS data sets must be mounted and their mount point known. The console system command **D OMVS,F** or the USS shell command **df** can be used to display all mounted USS data sets. If the Broker has never run as UID 0 or with READ access to the BPX.SUPERUSER profile, the datasets must be manually mounted with the mount locations configured and initialized as described below.
- The external links required by Universal Broker and UDM Manager to spawn select Universal Agent components must exist in a well-known location on the z/OS UNIX file system and be owned by UID 0. The external links must be created prior to Broker startup and identified in the configuration and component definition members described below.

Step 1	Stop the Universal Broker STC if it is running.
Step 2	Change the user profile **UBRUSR** UID value to 5001 with the following command: `ALTUSER UBRUSR OMVS(UID(5001))`

Step 3

Note

Some parts of this step are optional starting with Universal Agent 6.5.0.0. Running with access to all the profiles below is still supported in that release, but the only resource the Broker user must have READ access to is BPX.JOBNAME. Running without BPX.SUPERUSER and BPX.DAEMON access does limit the Universal Agent's feature set, but only as it applies to z/OS Unix group membership and configuring the Agent to switch user contexts without prior authentication.

Permit the user profile **UBRUSR** READ access to the required resource profiles with the following commands:

```
PE BPX.SUPERUSER CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
PE BPX.JOBNAME CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
SETR RACLIST(FACILITY) REFRESH
```

The user profile **UBRUSR** should already have READ access to **BPX.DAEMON** in the FACILITY class based on Universal Broker installation requirements prior to 4.2.0. If **UBRUSR** does not have READ access to **BPX.DAEMON**, the following commands will permit appropriate access:

```
PE BPX.DAEMON CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
SETR RACLIST(FACILITY) REFRESH
```

For detailed information regarding Universal Broker security requirements, see z/OS Configuration - Started Tasks.

Step 4

Universal Broker databases are maintained in USS HFS or zFS data sets. The database files have an owner attribute that is based on the UID value of the Universal Broker STC user profile. The database files, the root directory, and administration files must have their owner attribute changed from UID 0 to the new non-zero UID value 5001.

When running with UID 0 or with READ access to the BPX.SUPERUSER profile, the Universal Broker STC will dynamically mount the USS data sets in the **`/tmp`** directory. Assuming the USS data set names are **`UNV.UNVDB`** and **`UNV.UNVSPOOL`**, their mount point would be directory **`/tmp/UNV.UNVDB`** and **`/tmp/UNV.UNVSPOOL`**. If a different mount point must be used, its location can be changed with the MOUNT_POINT configuration option inside the `&HLQ.UNV.UNVCONF(UBRCFG00)` member. If necessary, adjust the commands below to the appropriate directory names.

If the Universal Broker STC has *never* run with UID 0 or with READ access to the BPX.SUPERUSER profile, the databases must be mounted manually as described in Universal Agent Database Configuration.

From the z/OS UNIX shell prompt, execute the following commands:

```
su
cd /tmp/UNV.UNVDB
chown -R 5001 *
chown 5001 .
chown 5001 .inited
cd /tmp/UNV.UNVSPOOL
chown -R 5001 *
chown 5001 .
chown 5001 .inited
exit
```

The first command, **su**, changes to the superuser ID. The user ID used to execute the above commands will need either a UID of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class. If the user ID has UID 0, the **su** command is not necessary.

The `.inited` file must exist in each directory and must be owned by the Universal Broker user. If the Broker has not run in this environment, it will be necessary to manually create this file, so its ownership can be set as instructed above. There are no content requirements for this file, so an empty file will suffice.

Step 5

Universal Broker spawns Universal Agent server components via external links that reside on the z/OS UNIX file system (USS). These links must point to names that match load modules installed into the SUNVLOAD load library.

UDM Manager also can spawn UCMD and USAP in response to an exec or execsap command, following the same external link approach used by Universal Broker. The Broker provides these links to UDM during registration.

If the Broker runs as UID 0 (or as a user with READ access to BPX.SUPERUSER), it will format and create each of these external links at start-up in the location specified by the TMP_DIRECTORY configuration option.

If the Broker does not run with superuser authority, each of these links must be created manually and their locations must be identified in the appropriate component definitions and configuration files.

To create external links for Universal Agent server components and have them recognized by Universal Broker:

- From the z/OS UNIX shell prompt, execute the following commands:

```
su
cd /tmp
ln -e UAGSRV ubroker.stc.uagsrv
ln -e UCMSRV ubroker.stc.ucmsrv
ln -e UCTSRV ubroker.stc.uctsrv
ln -e UDMSRV ubroker.stc.udmsrv
```

  The directory and name (for example, `ubroker.stc.uagsrv`) can be any desired, but the link must point to the name of the load module in `&HLQ.UNV.SUNVLOAD`.

  If the TMP_DIRECTORY configuration option - defined inside `&HLQ.UNV.UNVCONF(UBRCFG00)` - was changed from its default value of `/tmp`, place the external links in the location specified by the mount point value.
- Make the following changes to the component definition files, which are stored as members inside `&HLQ.UNV.UNVCOMP`. If necessary, replace `/tmp` with the configured tmp_directory location.
  - UAGSRV: Set start_command option to `/tmp/ubroker.stc.uagsrv` in UAGCMP00.
  - UCMSRV: Set start_command option to `/tmp/ubroker.stc.ucmsrv` in UCSCMP00.
  - UCTSRV: Set the start_command option to `/tmp/ubroker.stc.uctsrv` in UTSCMP00.
  - UDMSRV: Set the start command option to `/tmp/ubroker.stc.udmsrv` in UDSCMP00.

To create external links used by UDM to execute UCMD and USAP, execute the following z/OS Unix commands

```
su
cd /tmp
ln -e UCMD ubroker.stc.ucmd
ln -e USAP ubroker.stc.usap
```

Update the UCMD_PATH and USAP_PATH Universal Broker configuration options in `&HLQ.UNV.UNVCONF(UBRCFG00)` to point to the absolute path of those links.

If you expect to stop components from Universal Broker using UCTL, create a link for it as well (for example, In -e UCTL ubroker.stc.uctl) and update the UCTL_PATH option in `&HLQ.UNV.UNVCONF(UBRCFG00)`.

If you expect to launch started tasks (STCs) via UCMD using a COMMAND_TYPE value of **stc** or via the UDM **exec stc=** command, execute the following z/OS Unix commands to create an external link to the UCMD Server STC Command Processor, UCMSCPST.

```
su
cd /tmp
ln -e UCMSCPST ucmd.stc
```

Update the UCMSCPST_PATH UCMD Server configuration option in `&HLQ.UNV.UNVCONF(UCSCFG00)` to point to the absolute path of this new link.

Step 6

Start the Universal Broker STC.

Running Without BPX.SUPERUSER and BPX.DAEMON Access

Starting with the Universal Agent 6.5.0.0 release, running the Universal Broker STC with a user account without READ access to the BPX.SUPERUSER and BPX.DAEMON profiles enables more agent functionality than with previous releases.

The Universal Broker provided by the Universal Agent 6.5.0.0 release can execute Universal Server components (for example, UCMSRV and UDMSRV) in a specified user's context without requiring the Broker account to have access to these privileged resources. This simply requires that a valid password be provided for the user account, so that the Server component can authenticate the account.

While most agent functionality is available when executing this way, it does impose the following limitations:

- The **noauth** parameter supported by some Universal Access Control List entries may not be used. This parameter is used to perform user context switches without requiring user authentication. This behavior is only available if the Broker runs with UID 0 or has BPX.SUPERUSER access.
- Access to system resources that is granted to users via their group membership may need to be updated to specifically grant access to that user's account. Supplemental group information for the user will be set, but the process will be unable to set its effective group ID unless the user is also a member of the group to which the Broker user belongs.

In addition, the system log may contain an increased number of ICH408I messages reporting insufficient access to the BPX.SUPERUSER and BPX.DAEMON resources. This is expected behavior and is issued because some Universal Server components (UCMSRV in particular) issue a function call that checks for access to those resources. If the account requesting access does not have it, the ICH408I message is issued. Internally, the Server component continues, aware that access to the privileged resources is not available.

An ICH804I message issued by a Universal Server component may look like this:

```
ICH408I USER(UBRTRP   ) GROUP(UBRGRP   ) NAME(####################)
  BPX.SUPERUSER CL(FACILITY)
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
```

In this instance, the Broker was running with account UBRTRP, which does not have READ access to the BPX.SUEPRUSER resource of the FACILITY RACF class.

Before attempting to run the Universal Broker STC without access to these privileged resources, be sure to follow the manual configuration steps listed in # Converting Universal Broker User Profile to Non-Zero UID.

Converting Universal Enterprise Controller (UEC) User Profile to Non-Zero UID

The conversion steps assume the following:

- The UID value is being changed from 0 to 5002. If a UID value of 5002 does not work in your local environment, change all references to 5002 in the following steps to a unique, non-zero UID value suitable for your local environment. Note that the UID value must be unique among all user profiles.
- The UEC user profile name is **UECUSR**. If the UEC STC in your local environment uses a different user profile name, change all references to **UECUSR** in the following steps to the user profile name used in your local environment.
- The user ID used to execute the commands requires an OMVS segment
- To simplify database mounting, the user ID must have either UID 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class. To run without access to the BPX.SUPERUSER, simply mount, configure, and initialize databases as described in Universal Agent Database Configuration.
- The UEC HFS or zFS data set must be mounted and its mount point known. The console system command **D OMVS,F** or the USS shell command **df** can be used to display all mounted USS data sets.

Step 1	Stop the UEC STC if it is running.
Step 2	Change the user profile UECUSR UID value to 5002 with the following command: `ALTUSER UECUSR OMVS(UID(5002))`

Step 3	Optional: Permit the user profile **UECUSR** READ access to the required resource profiles with the following commands:
	
	`PE BPX.SUPERUSER CLASS(FACILITY) ID(UECUSR) ACCESS(READ)`
	`SETR RACLIST(FACILITY) REFRESH`

Step 4	UEC databases are maintained in a USS HFS or zFS data set. The database files have an owner attribute that is based on the UID value of the UEC STC user profile. The database files, the root directory, and administration files must have their owner attribute changed from UID 0 to the new non-zero UID value 5002.
	
	When running with UID 0 or with READ access to the BPX.SUPERUSER profile, the Universal Enterprise Controller STC will dynamically mount the USS data set in the **`/tmp`** directory. Assuming the USS data set name is **`UNV.UECDB`**, its mount point would be directory **`/tmp/UNV.UECDB`**. If a different mount point must be used, its location can be changed with the MOUNT_POINT configuration option inside the &`HLQ.UNV.UNVCONF(UECCFG00)` member. If necessary, adjust the following commands to reflect the correct mount point location.
	
	If the UEC STC has *never* run with UID 0 or with READ access to the BPX.SUPERUSER profile, the databases must be mounted manually as described in Universal Agent Database Configuration.
	
	From the z/OS UNIX shell prompt, execute the following commands:
	
	`su`
	`cd /tmp/UNV.UECDB`
	`chown -R 5002 *`
	`chown 5002 .`
	`chown 5002 .inited`
	`exit`
	
	The first command, **su**, changes to the superuser ID. The user ID used to execute the above commands will need either a UID of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class. If the user ID has UID 0, the **su** command is not necessary.
	
	The `.inited` file must exist in each directory and must be owned by the UEC user. If the UEC STC has not run in this environment, it will be necessary to manually create this file, so its ownership can be set as instructed above. There are no content requirements for this file, so an empty file will suffice.

Step 5	Start the UEC STC.

z/OS Installation - Data Set Inventory

- Types of Data Sets
- SMP/E Data Sets
- Non-SMP/E Data Sets

Types of Data Sets

As part of the Universal Agent for z/OS package installation, two types of data sets are allocated and cataloged:

- SMP/E data sets
- Non-SMP/E data sets

SMP/E Data Sets

The following table lists the SMP/E data sets — and their space requirements — that are allocated and cataloged as part of the Universal Agent for z/OS installation.

Depending on your installation choices, the data set high-level qualifiers may be different.

Data Set Name	Space (tracks)	Description
UNV.AUNVLOAD	6500	SMP/E distribution library for the product load library.
UNV.AUNVNLS	30	SMP/E distribution library for the product national language support library.
UNV.AUNVSAMP	30	SMP/E distribution library for the product sample library.
UNV.AUNVTMPL	60	SMP/E distribution library for configuration template files.
UNV.SMP.CSI	n/a	SMP/E CSI VSAM cluster for Universal Agent SMP/E zones.
UNV.SMP.CSI.DATA	75	SMP/E CSI VSAM data for Universal Agent SMP/E zones.
UNV.SMP.CSI.INDEX	15	SMP/E CSI VSAM index for Universal Agent SMP/E zones.
UNV.SMP.SMPLOG	30	SMP/E log file.
UNV.SMP.SMPLOGA	30	SMP/E backup log file.
UNV.SMP.SMPLTS	100	SMP/E target library for base versions of load modules using a SYSLIB allocation.
UNV.SMP.SMPMTS	30	SMP/E target library for macros existing only in the distribution libraries.
UNV.SMP.SMPPTS	4500	SMP/E temporary SYSMOD storage.
UNV.SMP.SMPSCDS	30	SMP/E zone backup library.
UNV.SMP.SMPSTS	30	SMP/E target library for source existing only in the distribution libraries.
UNV.SUNVLOAD	6500	SMP/E target library for the product load library.
UNV.SUNVNLS	30	SMP/E target library for product national language support library.
UNV.SUNVSAMP	30	SMP/E target library for the product sample library.
UNV.SUNVTMPL	60	SMP/E target library for configuration template file.

Non-SMP/E Data Sets

The following table lists the non-SMP/E data sets - and their space requirements - that are allocated and cataloged as part of the Universal Agent for z/OS package installation.

Depending on your installation choices, the data set high-level qualifiers may be different.

Data Set Name	Space (tracks)	Description

UNV.MDL	1	Universal Broker sequential trace data set allocation model.
UNV.UAG	1	Universal Automation Center Agent logging data set allocation model.
UNV.UAG.AGENT.LOG	n/a	Universal Automation Center Agent logging data set GDG base.
UNV.UCRDB	15	Universal Certificate database.
UNV.UECDB	4500	Universal Enterprise Controller HFS or zFS databases.
UNV.UNVCOMP	15	Universal Agent component definition library.
UNV.UNVCONF	15	Universal Agent configuration library.
UNV.UNVCREF	75	Universal Command Server command reference library.
UNV.UNVDB	150	Universal Broker HFS or zFS database.
UNV.UNVJSC	n/a	VSAM Job Submission Checkpoint (JSC) cluster.
UNV.UNVKSTR	75	Universal Broker Key Store data set.
UNV.UNVSPOOL	3000	Universal Agent HFS or zFS spool database.
UNV.UNVTRACE	150	Universal Broker PDS/E trace data set.
UNV.V6R4M0.INSTALL	30	Universal Agent package installation and maintenance JCL.

z/OS Installation - Licensing

- Licensing Universal Agent for z/OS Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

Licensing Universal Agent for z/OS Components

After Universal Agent for z/OS has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Connector for PeopleSoft
- Universal Connector for SAP

Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

Format

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: ***Indesca_Stonebranch_MVS_A1.txt***

- For Universal Command Manager, **Indesca** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.
- For Universal Data Mover Manager, **Infitran** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.

Sample

The following is a sample Universal Agent for z/OS product license file (for Universal Command Manager):

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Entering License Information

Enter each component's product license parameters into its configuration file in the **UNVCONF** product library:

Universal Command Manager configuration file	member **UCMCFG00**
Universal Data Mover Manager configuration file	member **UDMCFG00**
Universal Connector for PeopleSoft configuration file	member **UPPSFG00**

Universal Connector for SAP configuration file	member **USPCFG00**

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

If the Universal Broker STC is running, restart the Universal Broker STC to pick up the configuration file changes.

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it.

Stop Universal Broker	`STOP UBROKER`
Start Universal Broker	`START UBROKER[,UPARM='options']`

z/OS USS Installation

- Introduction
- Installation Procedures
- Detailed Information

Introduction

These pages describe the installation of Stonebranch, Inc.'s Universal Agent for z/OS UNIX System Services (USS) on a z/OS operating system. Unless otherwise specified, all references to Universal Agent for z/OS USS refer to version 7.2.x.

Note

Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Agent component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

Installation Procedures

All Universal Agent for z/OS USS components are provided in the Universal Agent for z/OS SMP/E package.

The z/OS components in that package must be installed before you can install the z/OS USS components. (See z/OS Installation Procedures for information on completing the z/OS installation.)

After the z/OS components have been installed, you can install the z/OS USS components (see SMPE Installation).

The installation consists of running a number of batch jobs. The output of these batch jobs should be kept until a proper installation has been verified.

Note

Please read z/OS USS Installation - Installation Requirements before installing Universal Agent for z/OS USS.

Detailed Information

The following pages provide detailed information for Configuration of z/OS USS Installation:

- Components
- Installation Requirements
- Distribution File
- SMPE Installation
- Customization
- Data Set Inventory
- Licensing

z/OS USS Installation - Installation Package

- Universal Agent for z/OS USS Components
- Component Compatibility

All Universal Agent for z/OS USS components are provided in the Universal Agent for z/OS SMP/E package.

Universal Agent for z/OS USS Components

The Universal Agent for z/OS package contains the following Universal Agent for z/OS USS Components:

- Universal Command Manager 7.2.x
- Universal Control Manager 7.2.x
- Universal Copy 7.2.x
- Universal Data Mover Manager 7.2.x
- Universal Encrypt 7.2.x
- Universal Event Monitor Manager 7.2.x
- Universal Message to Exit Code Translator 7.2.x
- Universal Query 7.2.x
- Universal Certificate 7.2.x
- Universal WTO 7.2.x

Component Compatibility

The following table identifies the compatibility of Universal Agent for z/OS USS components with previous component / product versions.

Component	Compatibility
Universal Command 7.2.x	Universal Command 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 7.2.x	Universal Control 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 7.2.x	Universal Data Mover 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.
Universal Encrypt 7.2.x	Universal Encrypt 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 7.2.x	Universal Broker 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Event Monitor 7.2.x	Universal Event Monitor 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

z/OS USS Installation - Installation Requirements

- Overview
- Data Set Space Requirements
- SMP/E
- Platform Requirements

Overview

Universal Agent for z/OS USS requires the following software releases:

- z/OS 2.2, 2.3, or 2.4.
- IBM Language Environment for z/OS 2.2, 2.3, or 2.4.
- IBM Communication Server for z/OS 2.2, 2.3, or 2.4.
- SMP/E 3.5 or above.
- Minimum 200 cylinders of DASD and 81M bytes on a z/OS UNIX file system.

The user identifier used to execute the installation jobs must meet the following requirements:

1. User ID must have a properly defined OMVS segment.
2. User ID must have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Important

All Universal Agent for z/OS USS components are provided in the Universal Agent for z/OS SMP/E package. The z/OS components must be installed before the z/OS USS components. See z/OS Installation for information on completing the z/OS installation.

Data Set Space Requirements

As part of the Universal Agent for z/OS USS installation, a number of SMP/E data sets are allocated and cataloged and a number of z/OS UNIX directories are created.

The space requirements for these data sets and directories are listed in z/OS USS Installation - Data Set Inventory.

SMP/E

Universal Agent for z/OS USS is installed using SMP/E. The components are installed in the Universal Agent SMP/E CSI.

The following table identifies the SMP/E FMIDs for the Universal Agent for z/OS USS components.

Product	FMID	SMP/E Requisites
USS Universal Common 7.2.x	UUNV720	No prerequisites. Supersedes and deletes FMID UUNV320, UUNV410, UUNV420, UUNV430, UUNV510, UUNV520, UUNV620, UUNV630, UUNV640, UUNV650, UUNV660, UUNV670, UUNV680, UUNV690, UUNV700, and UUNV710.
USS Universal Broker Utilities 7.2.x	UUBR720	UUNV710 and TUBR700 are prerequisites. Supersedes and deletes FMID UUBR320, UUBR410, UUBR420, UUBR430, UUBR510, UUBR520, UUBR620, UUBR630, UUBR640, UUBR650, UUBR660, UUBR670, UUBR680, UUBR690, UUBR700, and UUBR710.
USS Universal Command Manager 7.2.x	UUCM720	UUNV710 and TUBR700 are prerequisites. Supersedes and deletes FMID UUCM320, UUCM410, UUCM420, UUCM430, UUCM510, UUCM520, UUCM620, UUCM6630, UUCM640, UUCM650, UUCM660, UUCM670, UUCM680, UUCM690, UUCM700, and UUCM710.
USS Universal Data Mover Manager 7.2.x	UUDM720	UUNV710 and TUBR700 are prerequisites. Supersedes and deletes FMID UUDM320, UUDM410, UUDM420, UUDM430, UUDM510, UUDM520, UUDM620, UUDM630, UUDM640, UUDM650, UUDM660, UUDM670, UUDM680, UUDM690, UUDM700, and UUDM710.
USS Universal Utilities 7.2.x	UUTL720	UUNV710 is a prerequisite. Supersedes and deletes FMID UUTL320, UUTL410, UUTL420, UUTL430, UUTL510, UUTL520, UUTL620, UUTL630, UUTL640, UUTL650, UUTL660, UUTL670, UUTL680, UUTL690, UUTL700, and UUTL710.
USS Universal Event Monitor Manager 7.2.x	UUEM720	UUNV710 and TUBR700 are prerequisites. Supersedes and deletes FMID UUEM320, UUEM410, UUEM420, UUEM430, UUEM510, UUEM520, UUEM620, UUEM630, UUEM640, UUEM650, UUEM660, UUEM670, UUEM680, UUEM690, UUEM700, and UUEM710.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

z/OS USS Installation - Distribution File

Universal Agent for z/OS USS - Product Distribution File

The Universal Agent for z/OS USS installation files are distributed with the Universal Agent for z/OS distribution file.

z/OS USS Installation - SMPE Installation

Universal Agent for z/OS USS - SMP/E Installation

The installation steps describe how to perform the SMP/E install of Universal Agent for z/OS USS in a step-by-step process.

Note

Installation of z/OS USS is dependent upon completion of the z/OS package installation, and must be installed in the same CSI as the z/OS package.

Four different installation processes are provided. Which installation process used depends on the installation environment.

z/OS USS Installation - New Install, New CSI	Install any Universal Agent package from Stonebranch, Inc. for the first time, or installing a Universal Agent for z/OS USS package in a new SMP/E CSI.
z/OS USS Installation - 7.1.x Upgrade, Existing CSI	Upgrade a Universal Agent 7.1.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 7.1.x package.
z/OS USS Installation - 7.0.x Upgrade, Existing CSI	Upgrade a Universal Agent 7.0.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 7.0.x package.
z/OS USS Installation - 6.9.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.9.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.9.x package.
z/OS USS Installation - 6.8.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.8.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.8.x package.
z/OS USS Installation - 6.7.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.7.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.7.x package.
z/OS USS Installation - 6.6.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.6.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.6.x package.
z/OS USS Installation - 6.5.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.5.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.5.x package.
z/OS USS Installation - 6.4.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.4.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.4.x package.
z/OS USS Installation - 6.3.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.3.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.3.x package.
z/OS USS Installation - 6.2.x Upgrade, Existing CSI	Upgrade a Universal Agent 6.2.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 6.2.x package.
z/OS USS Installation - 5.2.0 Upgrade, Existing CSI	Upgrade a Universal Agent 5.2.0 for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Agent 5.2.0 package.
z/OS USS Installation - 5.1.0 Upgrade, Existing CSI	Upgrade a Workload Automation 5.1.0 for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Workload Automation 5.1.0 package.
z/OS USS Installation - 4.x Upgrade, Existing CSI	Upgrade a Stonebranch Solutions 4.x for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Stonebranch Solutions 4.x package.
z/OS USS Installation - 3.2.0 Upgrade, Existing CSI	Upgrade a Universal Products 3.2.0 for z/OS USS package install. In this case, the Universal Agent 7.2.x for z/OS USS package is installed into an SMP/E CSI that contains a Universal Products 3.2.0 package.

z/OS USS Installation - New Install, New CSI

New Install, New CSI

The New Install, New CSI installation process describes how to install the Universal Agent for z/OS USS package in a newly allocated SMP/E CSI.

Use this installation process for either of these environments:

- Installing a Universal Agent for z/OS USS package for the first time.
- Installing a Universal Agent for z/OS USS package in a different SMP/E CSI than other Universal Agent components.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN01**. The JCL allocates the SMP/E target and distribution data sets, creates the z/OS UNIX directories, and defines the DDDEFs in the SMP/E zones. All steps must end with a return code 0. Step **ALLOCUSS** creates the z/OS UNIX directories required by the USS components. The path prefix where the directories are created was defined when the **#SETUP** member of the **INSTALL** library was modified and run as part of the z/OS installation. IBM recommends the path prefix to be **/usr/lpp**. Review the path prefix used in the **ALLOCUSS** step PARM value to be sure it meets local requirements. The z/OS UNIX directory where the Universal Agent directories are created must be mounted in read/write mode. The user ID used to run this job must have write access to the directory. If the z/OS UNIX directories already have been created, the **ALLOCUSS** step will indicate this in its report and end successfully.
Step 2	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 3	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step **APYFMID** must end with a condition code of 0. Step **APYPTFS** is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname SMPOUT. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 4	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step **ACCFMID** must end with a condition code of 0. Step **ACCPTFS** is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
Step 5	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 7.1.x Upgrade, Existing CSI

Universal Agent 7.1.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 7.1.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 7.1.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 7.1.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 7.1.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 7.0.x Upgrade, Existing CSI

Universal Agent 7.0.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 7.0.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 7.0.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 7.0.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 7.0.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.9.x Upgrade, Existing CSI

Universal Agent 6.9.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.9.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.9.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.9.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.9.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.8.x Upgrade, Existing CSI

Universal Agent 6.8.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.8.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.8.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.8.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.8.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.7.x Upgrade, Existing CSI

Universal Agent 6.7.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.7.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.7.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.7.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.7.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.6.x Upgrade, Existing CSI

Universal Agent 6.6.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.6.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.6.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.6.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.6.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.5.x Upgrade, Existing CSI

Universal Agent 6.5.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.5.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.5.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.5.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.5.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.4.x Upgrade, Existing CSI

Universal Agent 6.4.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.4.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.4.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.4.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.4.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.3.x Upgrade, Existing CSI

Universal Agent 6.3.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.3.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.3.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.3.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.3.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 6.2.x Upgrade, Existing CSI

Universal Agent 6.2.x for z/OS USS Upgrade, Existing CSI

The Universal Agent 6.2.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 6.2.x package in an existing SMP/E CSI.

Note

Before the Universal Agent 6.2.x for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 6.2.x for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 5.2.0 Upgrade, Existing CSI

Universal Agent 5.2.0 for z/OS USS Upgrade, Existing CSI

The Universal Agent 5.2.0 for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Agent 5.2.0 package in an existing SMP/E CSI.

Note

Before the Universal Agent 5.2.0 for z/OS USS package can be upgraded, you first must upgrade the Universal Agent 5.2.0 for z/OS package to Universal Agent 7.2.x.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 5.1.0 Upgrade, Existing CSI

Workload Automation 5.1.0 for z/OS USS Upgrade, Existing CSI

The Workload Automation 5.1.0 for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Workload Automation 5.1.0 package in an existing SMP/E CSI.

Note

Before the Workload Automation 5.1.0 for z/OS USS package can be upgraded, you first must upgrade the Workload Automation 5.1.0 for z/OS package to Universal Agent 5.2.0.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 4.x Upgrade, Existing CSI

Stonebranch Solutions 4.x for z/OS USS Upgrade, Existing CSI

The Stonebranch Solutions 4.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Stonebranch Solutions 4.x package in an existing SMP/E CSI.

Note

Before the Stonebranch Solutions 4.x for z/OS USS package can be upgraded, you first must upgrade the Stonebranch Solutions 4.x for z/OS package to Universal Agent 5.2.0.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - 3.2.0 Upgrade, Existing CSI

Universal Products 3.2.0 for z/OS USS Upgrade, Existing CSI

The Universal Products 3.2.0 for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Products 3.2.0 package in an existing SMP/E CSI.

Note

Before the Universal Products 3.2.0 for z/OS USS package can be upgraded, you first must upgrade the Universal Products 3.2.0 for z/OS package Universal Agent 5.2.0.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Agent **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

Step 1	Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
Step 2	Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions: • Step ends with condition code 0. • Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**. The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.
Step 3	Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions: • Step ends with condition code 0 or 4. • Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
Step 4	Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

z/OS USS Installation - Customization

- Overview
- Universal Command Manager for z/OS USS Customization
  - Configuration
- Universal Control Manager for z/OS USS Customization
  - Configuration
- Universal Data Mover Manager for z/OS USS Customization
  - Configuration
- Universal Event Monitor Manager for z/OS USS Customization
  - Configuration
- Universal Query for z/OS USS Customization
  - Configuration

Overview

The product executable files intended for command line use are located in the directory **#USSPRE/universal/bin**, where **#USSPRE** is the path prefix where the USS component directories were created. This directory must be added to the PATH environment variable for intended users of the executable files.

(For information on applying product licenses to installed Universal Agent for z/OS USS components, see z/OS USS Installation - Licensing.)

Universal Command Manager for z/OS USS Customization

Configuration

USS Universal Command Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UCMCFGU0** of the **UNVCONF** library.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Manager.

Universal Control Manager for z/OS USS Customization

Configuration

USS Universal Control Manager for z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UCTCFGU0** of the **UNVCONF** library.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Manager.

Universal Data Mover Manager for z/OS USS Customization

Configuration

USS Universal Data Mover Manager for z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UDMCFGU(** of the **UNVCONF** library.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring Universal Data Mover.

Universal Event Monitor Manager for z/OS USS Customization

Configuration

USS Universal Event Monitor Manager for z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UEMCFG U0** of the **UNVCONF** library.

See the Universal Event Monitor 7.2.x Reference Guide for details on configuring Universal Event Monitor Manager.

Universal Query for z/OS USS Customization

Configuration

USS Universal Query z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UQRCFGU0** of the **UNVCONF** library.

See Universal Query for details on configuring Universal Query.

z/OS USS Installation - Data Set Inventory

SMP/E Data Sets

A number of SMP/E data sets are allocated and cataloged as part of the Universal Agent for z/OS USS installation.

The following table lists the data sets and their space requirements.

Depending on your installation choices, the data set high-level qualifiers may be different.

Data Set Name	Space (tracks)	Description
UNV.AUNVHBIN	3000	SMP/E distribution library for z/OS UNIX executables.
UNV.AUNVHNLS	30	SMP/E distribution library for z/OS UNIX product national language support files.

z/OS UNIX Directories

A number of z/OS UNIX directories are created as part of the Universal Agent for z/OS USS installation.

The following table lists the directories and their space requirements.

Directory Name	Space (MB)	Description
universal/bin/IBM	80	SMP/E target directory for the SUNVHBIN DDDEF.
universal/bin	0	USS Universal Agent program files.
universal/nls/IBM	0.5	SMP/E target directory for the SUNVHNLS DDDEF.
universal/nls	0	USS Universal Agent national language support files.
universal/ucmdmgr	0	Universal Command Manager installation directory.
universal/uctlmgr	0	Universal Control Manager installation directory.
universal/udmmgr	0	Universal Data Mover Manager installation directory.
universal/uemmgr	0	Universal Event Monitor Manager installation directory.
universal/uquery	0	Universal Query installation directory.

z/OS USS Installation - Licensing

- Licensing Universal Agent for z/OS USS Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

Licensing Universal Agent for z/OS USS Components

After Universal Agent for z/OS USS has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager

Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

Note

Product license information for Universal Agent for z/OS USS components are contained in the Universal Agent for z/OS product license files.

Format

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: ***Indesca_Stonebranch_MVS_A1.txt***

- For Universal Command Manager, **Indesca** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.
- For Universal Data Mover Manager, **Infitran** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.

Sample

The following is a sample Universal Agent for z/OS product license file (for Universal Command Manager):

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Entering License Information

Enter each component's product license parameters into its configuration file in the **UNVCONF** product library:

- Universal Command Manager configuration file: member **UCMCFGU0**.
- Universal Data Mover Manager configuration file: member **UDMCFGU0**.

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

If the Universal Broker STC is running, restart the Universal Broker STC to pick up the configuration file changes.

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it.

Stop Universal Broker	`STOP UBROKER`
Start Universal Broker	`START UBROKER[,UPARM='options']`

Windows Installation

- Introduction
- Installation Packages
  - Distribution File Format
- Universal Agent Installation Methods
  - System Installation
  - User Mode Installation
- Pre-Installation Guidelines
  - System Installation
  - User Mode Installation
- Installation Paths
  - System Installation
  - User Mode Installation
- Installation Summary
  - System Installation
  - User Mode Installation
- Windows Installer
  - Windows Installer Interfaces
  - Determining Whether a Universal Agent Install Will Require a System Reboot
  - Windows Installer Package File Locations
- Detailed Information

Introduction

These pages provide information on the installation of Stonebranch, Inc.'s Universal Agent on Microsoft Windows operating systems. Unless otherwise specified, all references to Universal Agent for Windows refer to one of the 7.2.x versions (for example: 7.2.0.0).

Installation Packages

Universal Agent is installed with one required installation package and three optional installation packages.

To obtain a Universal Agent package, you must download the corresponding product distribution file from the Stonebranch Customer Portal.

Note

A customer user name and password - provided by Stonebranch, Inc. - are required to access this area.

The following table lists the packages available, their supported platforms, distribution file names, and whether or not the package is required.

Package	Distribution File	Required
Universal Agent (system installation)	**sb-7.2.0.*LEVEL*-windows-x64.exe**	Yes
Universal Agent (user mode installation)	**sb-7.2.0.LEVEL-windows-x64-um.exe**	No
Universal Enterprise Controller	**sb-UECtlr-6.3.0.*LEVEL*-windows-i386.exe (32-bit)**	No
Universal Enterprise Controller Client Applications	**sb-UEClient-6.3.0.*LEVEL*-windows-i386.exe (32-bit)**	No
Universal Agent for SOA	**sb-soa-6.7.0.*LEVEL*-windows-x64.exe**	No

The word *LEVEL* in each distribution file name is a numeric value representing the product maintenance level contained in the distribution file.

Distribution File Format

The product distribution files are in a Windows self-extracting archive file format.

The names of the distribution files have the following format:

PROD-*VRML*-*OS*-*HW*.exe

- *PROD* is the package content
- *VRML* is the version, release, modification, and maintenance levels (for example, 7.2.0.0)
- *OS* is the name of the operating system supported

- *HW* is the hardware type supported

Universal Agent Installation Methods

There are two installation methods available for Universal Agent for Windows packages:

1. The first is a traditional Windows Installer-based installation, referred to here as a system installation.
2. The second is referred to as a user mode installation, which is a lightweight, fully-functional install that allows multiple Universal Agents to reside on a single system.

Detailed information for each method can be found on the following pages.

- System Installation
- User Mode Installation

System Installation

The system installation method uses a traditional Windows Installer-based package. This installation method "assumes" that it is the only Universal Agent instance on a system, and has the following distinguishing characteristics:

- The application is listed in Programs and Features, under the Windows Control Panel.
- Application files default to traditional install locations (for example, C:\Program Files).
- Universal Manager and Utility components are installed in the Windows System32 directory.
- Configuration and component definition files reside in in common, well-known locations (for example, C:\ProgramData).
- The installation contains the logic necessary to create a new account for running the Universal Broker service.
- The Universal Broker service name is not configurable. Its display name is `Universal Broker`, its registered service name is `UBrokerService`, and its Application event log message source is `ubroker`.
- The install contains logic to support upgrades from any prior release.
- Removal requires a proper Uninstall, via the Windows Control Panel.
- The installation is compatible with enterprise deployment tools (for example, SMS).
- The installation may install system libraries that the Universal Agent requires.
- The installation can automatically configure the Agent to simplify integration with other Stonebranch applications.

User Mode Installation

The user mode install is a lightweight installation method that results in a fully-functional Universal Agent, minus the Windows integration features that the system install offers. However, this install method permits several Universal Agents to reside on a single system, whereas the system install is limited to a single instance.

A user mode install can reside on a system that already has a system install, but an existing system install is not required.

A user mode install has the following distinguishing characteristics:

- Application files can reside in any location.
- Resources the Agent requires such as configuration and component definition files reside under the install location.
- Universal Manager and Utility components also reside under the install directory, in the `.\Sys32\bin` folder. To ensure that Managers and Utilities register with the correct local Broker, the `SB_ConfigSetup.bat` script is provided to properly configure the environment.
- By default, a user mode install provides a single working directory for Universal Server components, `.\UAHome`. A user mode install can still use separate directories for each Server component (as is done for the system install), but doing so requires post-install updates to configuration and component definition files.
- A user mode install can configure the Universal Broker service to run as an Administrative user account, but that account must already exist.
- The Universal Broker service name is configurable, and must be unique. The install creates the service and registers its Application event log as the service's display name.
- A user mode install contains no upgrade logic. Consequently, all user mode installs should be installed in "clean" locations.
- The user mode install does **not** create an entry in Programs and Features. To remove a user mode install, you 1) run the provided `ubrsvc.bat` script with the -remove option to delete the Broker service and unregister its Application event log message source, and 2) remove the install directory from the file system.

Note

A user mode install does not provide any required system libraries. All prerequisites must exist to run the Universal Agent in user mode. If the install cannot find the appropriate Microsoft Visual C++ Runtime and the Microsoft Universal C Runtime, the install will fail with an error.

Pre-Installation Guidelines

System Installation

Before starting the Universal Agent for Windows system installation, it is recommended that you close all other Windows programs. Active programs may be using system files that the Universal Agent installation may update.

If a system file is in use during the installation, it is placed in a temporary location. A reboot is required to free the file and move it from the temporary location to its final destination. Closing all programs before the installation reduces the possibility that a file will be in use and that you will have to reboot your computer.

You must be able to write to the directory from which the installation is launched.

External Runtime Dependencies

Additionally, some Universal Agent components may rely on vendor-supplied resources whose runtime dependencies are not known to the Agent before execution. One such component is the Universal Connector for SAP (USAP), which relies on libraries provided by SAP. If the runtime dependencies for those libraries are not available on your Agent system, attempts to run USAP will fail. Where such dependencies exist, use information provided by the vendor to determine runtime requirements. For example, to review runtime requirements for the NetWeaver RFC SDK version 750, refer to the information on this page.

User Mode Installation

User mode installations do not update shared system files, so it is not necessary to close all other applications first. However, we do suggest that you close the Windows Services application and the Windows Event Viewer, if open. A user mode installation will create the Universal Broker service and register its message catalog as an Application event log message source. File contention/sharing issues may occur if the Services and/or Event Viewer is open during Universal Broker service creation and configuration.

Note

Do not install a new user mode installation on top of an existing user mode installation. The installation contains no upgrade logic, so doing so can leave the installation in an indeterminate state.

External Runtime Dependencies

Additionally, some Universal Agent components may rely on vendor-supplied resources whose runtime dependencies are not known to the Agent before execution. One such component is the Universal Connector for SAP (USAP), which relies on libraries provided by SAP. If the runtime dependencies for those libraries are not available on your Agent system, attempts to run USAP will fail. Where such dependencies exist, use information provided by the vendor to determine runtime requirements. For example, to review runtime requirements for the NetWeaver RFC SDK version 750, refer to the information on this page.

Installation Paths

System Installation

The default installation path for most packages is:

- "**`C:\Program Files`**"

For the remaining 32-bit packages, the default installation path is:

- "**`C:\Program Files (x86)`**"

User Mode Installation

The default installation path for user mode installs is **`%TEMP%\universal`**. However, you can install to any location.

Installation Summary

System Installation

Step 1	Download the distribution file from the Stonebranch Customer Portal.
Step 2	Log on to Windows using an Administrative account or an account that can be elevated to Administrative authority.
Step 3	Exit all running programs to reduce the likelihood of a reboot.
Step 4	Execute the installation file.
Step 5	Follow installation instructions: • Installing a Universal Agent System Install via the Graphical Interface • Installing a Universal Agent System Install via the Command Line Interface

User Mode Installation

Step 1	Download the distribution file from the Stonebranch Customer Portal.
Step 2	Close the Windows Services application and the Windows Event Viewer.
Step 3	Execute the installation file, providing elevated credentials when/if asked to do so.
Step 4	Follow the instructions for your desired installation method: • Installing a Universal Agent User Mode Install via the Graphical Interface • Installing a Universal Agent User Mode Install via the Command Line Interface

Windows Installer

Note

All information related to Windows Installer applies only to the system installation. The user mode installation does not provide or depend on the Windows Installer.

The installation of each Universal Agent for Windows package requires Windows Installer (**msiexec.exe**) from Microsoft. Windows Installer is a service application that is a standard component of Windows operating systems. (It also can be obtained from Microsoft's website.)

Each Universal Agent for Windows package provides version **3.1.4000.2435** of Windows Installer.

Windows Installer Interfaces

Windows Installer provides two interfaces that you can use to install Universal Agent:

- Graphical user interface
- Command line

Determining Whether a Universal Agent Install Will Require a System Reboot

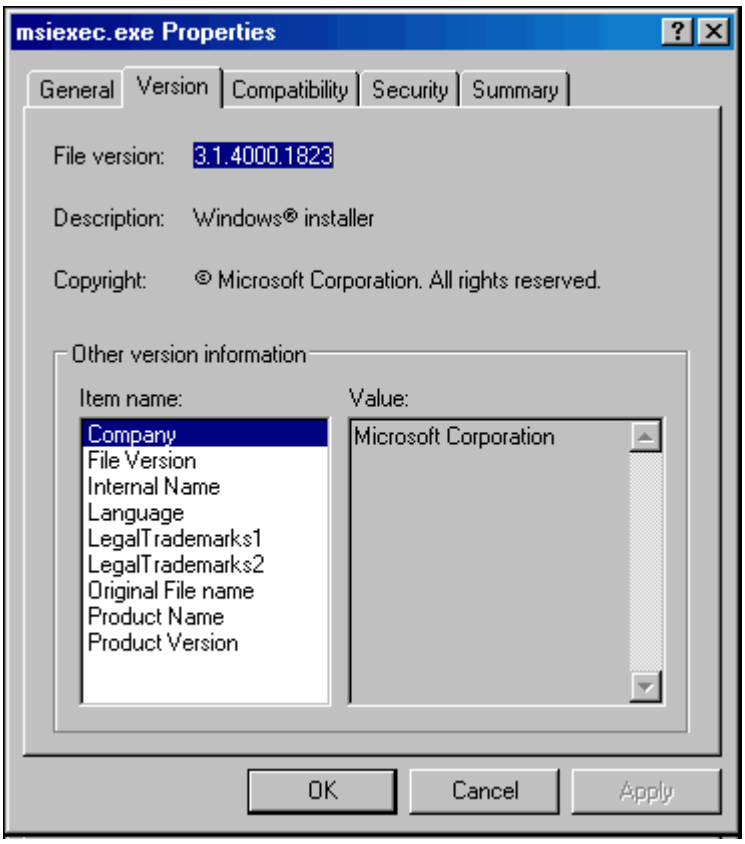
If Windows Installer is not present on your Windows operating system, the Universal Agent package being installed will install it before any of its own files are copied.

If Windows Installer is present on your Windows operating system, but it is a version prior to **3.1.4000.2435**, the installation will upgrade it. It may be necessary to reboot your system in this situation.

Recent versions of Microsoft Windows and Microsoft Server come with a version of Windows Installer that is much newer than the one provided by the Universal Agent install. This reduces the likelihood that a reboot will be required following an Agent install.

Follow the steps below to determine whether the Universal Agent install will install or upgrade the Windows Installer.

Step 1	From the Windows File Explorer, navigate to the System32 directory (for example, **C:\Windows\System32)**.
Step 2	Search for file **msiexec.exe** (Windows Installer). • If the file exists, continue to the Step 3. • If the file does not exist, Windows Installer is not installed on your system. The Universal Agent installation will install it.
Step 3	Right-click on the **msiexec.exe** file name to display a pop-up menu.
Step 4	Click **Properties** to display the Properties dialog for **msiexec.exe**.

Step 5	Click the **Version** tab. File Version: identifies the currently installed version of Windows Installer (see the following figure). If the version is prior to **3.1.4000.2435**, the package installation will upgrade it.  (In this figure, the installed Windows Installer is identified as version **3.1.4000.1823**. Since this is the earlier version than the version provided by the package installation, it will be upgraded.)

Rebooting the System

If Windows Installer is installed or upgraded during the installation of the package, a reboot of the system is required.

The installation provides an option to either:

- Reboot after the installation is complete.
- Reboot immediately. The installation will resume automatically after Windows restarts.

If Windows Installer version **3.1.4000.2435** or above already exists on your system, it is not upgraded, and no reboot is required.

Windows Installer Package File Locations

When you install a Universal Agent for Windows package, the Windows Installer package file (***.msi**) is extracted to the following locations:

Package	Location
Universal Agent for Windows package (64-bit)	`<LocalAppData>\UniversalAgentx64\<packagecode>\UCmdx64.msi`
Universal Enterprise Controller for Windows package	`<LocalAppData>\UniversalEnterpriseController\<packagecode>\UECtlr.msi`
UEC Client Applications package	`<LocalAppData>\UEClient\<packagecode>\UEClient.msi`
Universal Agent for SOA for Windows Package	`<LocalAppData>\UniversalAgentforSOAx64\<packagecode>\ Stonebranch Universal Agent for SOA (x64).msi`

In these paths:

<LocalAppData>	User's local Application Data folder. For example: If the installation was performed by the built-in Administrator account, **<LocalAppData>** would expand - by default - to: • `C:\Users\Administrator\AppData\Local`.
<packagecode>	Universally Unique Identifier (UUID) in the format {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}, where X is a hexadecimal character in the range 0 (zero) - F. For example: • {3B10285A-9602-4DC9-B0A5-4D701BEB5225}.

Detailed Information

The following pages provide detailed information for Windows Installation:

- Universal Agent for Windows System Installation
- Universal Agent for Windows User Mode Installation
- Universal Enterprise Controller for Windows Installation
- UEC Client Applications Installation
- Universal Agent for SOA for Windows Installation
- Licensing

Universal Agent for Windows System Installation

Introduction

The following information is provided for the installation of Universal Agent for Windows:

- Installation Package
- Installation Requirements
- Installation Procedures
- File Inventory Lists

(For licensing information, see Windows Installation - Licensing.)

Universal Agent for Windows - System Installation Package

Components

The Universal Agent for Windows System Installation package includes the following product components:

- Universal Broker 7.2.x
- Universal Automation Center Agent 7.2.x
- Universal Certificate 7.2.x
- Universal Command Manager and Server 7.2.x
- Universal Connector for SAP 7.2.x
- Universal Connector for PeopleSoft 7.2.x
- Universal Control Manager and Server 7.2.x
- Universal Controller Command Line Interface (CLI) 7.2.x
- Universal Data Mover Manager and Server 7.2.x
- Universal Encrypt 7.2.x
- Universal Event Log Dump 7.2.x
- Universal Event Monitor Manager and Server 7.2.x
- Universal Message Service (OMS) 7.2.x
- Universal Message to Exit Code Translator 7.2.x
- Universal Query 7.2.x
- Python 3.7 Distribution for Universal Agent (optional)

Note

Universal Enterprise Controller 6.3.x, Universal Enterprise Controller 6.3.x Client Applications, and Universal Agent for SOA 6.7.x are packaged separately (see Universal Enterprise Controller for Windows - Installation Package, UEC Client Applications - Installation Package, and Universal Agent for SOA for Windows - Installation Package).

Component Compatibility

The following table identifies the compatibility of Universal Agent for Windows components with previous component / product versions.

Component	Compatibility
Universal Broker 7.2.x	Universal Agent / Workload Automation / Stonebranch Solutions / Universal Products releases 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Command 7.2.x	Universal Command 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Control 7.2.x	Universal Control 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Data Mover 7.2.x	Universal Data Mover 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1 and 3.1.0.
Universal Query 7.2.x	Universal Broker 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Event Monitor 7.2.x	Universal Event Monitor 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

Universal Agent for Windows - System Installation Requirements

- Windows Versions
- Additional Requirements
- Platform Requirements
- Third-Party Requirements
- Installation Account
- Universal Broker Service
  - UBrokerService: The Default Universal Broker Service Account
  - Using a Windows Domain Account to Execute Universal Broker
- Spool Directory
  - Universal Command Server
  - UAG Cache

Windows Versions

To install Universal Agent for Windows, you must have one of the following versions of Windows:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

Additional Requirements

- An account with administrative privileges.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 325 megabytes of disk space.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Third-Party Requirements

Some Universal Agent components may rely on vendor-supplied resources whose runtime dependencies are not known to the Agent before execution. One such component is the Universal Connector for SAP (USAP), which relies on libraries provided by SAP. If the runtime dependencies for those libraries are not available on your Agent system, attempts to run USAP will fail. Where such dependencies exist, use information provided by the vendor to determine runtime requirements. For example, to review runtime requirements for the NetWeaver RFC SDK version 750, refer to the information on this page.

Installation Account

The account used to install the package must have administrative privileges.

Universal Broker Service

The Universal Broker service runs either with an Administrative account or with the Local System account.

If you are using an Administrative account, the account must have the following privileges:

- Act as part of the operating system
- Adjust memory quotas for a process
- Bypass traverse checking
- Debug programs
- Deny log on as batch job
- Deny log on locally

- Deny log on through Terminal Services
- Log on as a service
- Impersonate a client after authentication
- Increase scheduling priority
- Replace a process level token
- Take ownership of files and other objects

UBrokerService: The Default Universal Broker Service Account

The Universal Agent for Windows installation will create an account with the privileges listed above, if the account does not already exist. The name of this account is **`UBrokerService`**. A default password provided for the account can be obtained by contacting Stonebranch support. We strongly suggest that you set this password at install time via the installation dialogs or via the **BROKERPWD** command line option.

Note

If the UBrokerService account already exists, its password cannot be changed from the install.

See Installing a Universal Agent System Install via the Graphical Interface for detailed information about setting up the Universal Broker service account at install time.

Using a Windows Domain Account to Execute Universal Broker

The Universal Broker Service may be configured to execute with an existing Windows domain account, provided that account has been granted the privileges listed in this section, above, on the local system and the account has the file system permissions described in Service Security.

Note

Depending on your environment, it may be necessary to add this account to the Domain Admins group. This will ensure the account has sufficient access to domain resources and is recognized as a true administrative account on all domain member systems that run the Universal Broker service as that account.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Agent components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with Manager Fault Tolerance enabled.
- Configuration information for Universal Agent components, when a local Universal Broker is operating in managed mode.

The default location for the spooled standard I/O files is **`.\Universal\spool\ucmdsrv`**.

The default location for the other database files is **`.\Universal\spool\ubroker`**.

Universal Command Server

Location

The spool directory must reside on a local device. It cannot reside on any network device, including network drives that may be mapped to a local drive. By default, the spool files are located in directory **`.\Universal\spool\ucmdsrv`**.

Space

You must have approximately 50 megabytes of disk space for the installation.

The amount of disk space required for the spool directory depend on the following factors:

- Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
- When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
- Average size of the user processes standard input, output, and error files. Keep in mind that spooling is not a feature for file transfer purposes. File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = required disk space.`

For example, if the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. This feature can be enabled through the Universal Configuration Manager.

For more information on the Manager Fault Tolerant feature, spooling of redirected standard I/O files, and Universal Configuration Manager, see the Universal Agent 7.2.x User Guide.

Security

Universal Broker and Universal Command Server require read/write access to the spool directory. No other Universal Agent components access the spool directory directly.

UAG Cache

UAG cache is used by Universal Automation Center Agent (UAG) for storing standard I/O files.

Space

Cache files are located, by default, in directory **`.\Universal\UagSrv\cache`**.

Cache files are created for each job that is run by Universal Agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by Universal Controller. You can configure the number of days that files remain in the cache using the Agent Cache Retention Period in Days Universal Controller system property.

The amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Controller daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.`

For example:

If the files are purged every 7 days, and you run 1200 jobs on that agent server daily, and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

UAG automatically redirects the standard output and standard error files to the cache directory with no required user input.

Security

Universal Automation Center Agent (UAG) requires read/write access to the UAG cache directory. No other Universal Agent components access the cache directory. No general user access is required.

Universal Agent for Windows - System Installation Procedures

Installation Procedures

The following procedures are provided for the installation and modification of Universal Agent for Windows:

- Installing a Universal Agent System Install via the Graphical Interface
- Modifying a Universal Agent System Install via the Graphical Interface
- Installing a Universal Agent System Install via the Command Line Interface
- Modifying a Universal Agent System Install via the Command Line Interface

Note

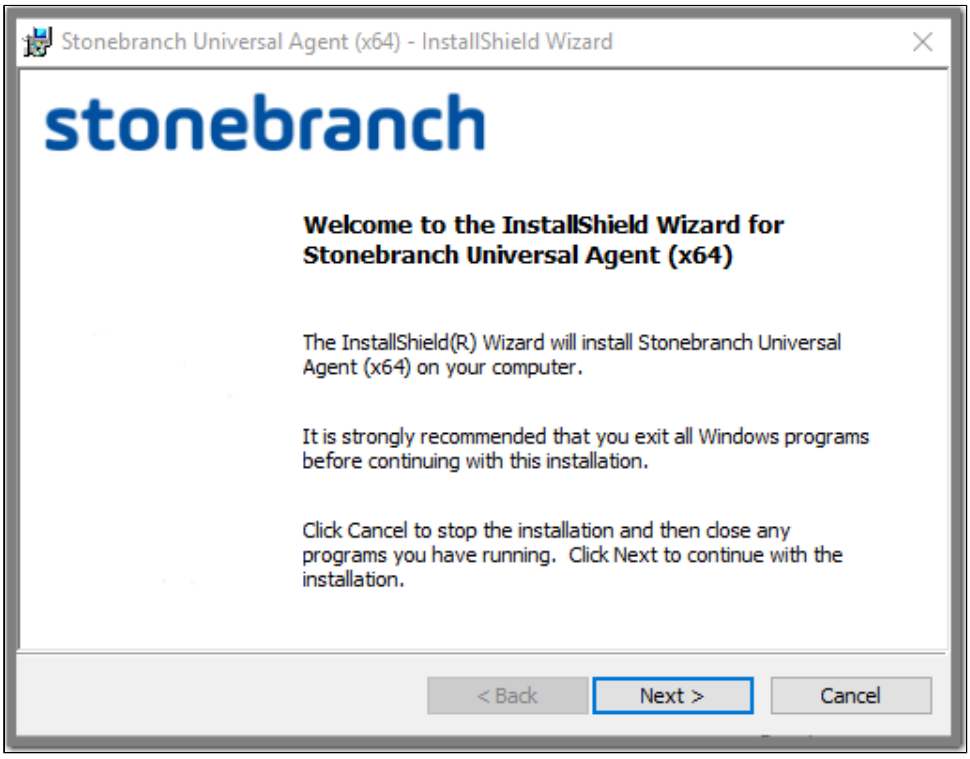
Modifying an Agent installation refers to the adding / removing of Agent components, repairing a corrupted installation, or removing an installation. To change the installed version of an Agent, see Upgrading Universal Agent and Applying Maintenance to Universal Agent.

Installing a Universal Agent System Install via the Graphical Interface

- Installing a Universal Agent for Windows System Install via the Windows Installer Graphical Interface
- Windows Installer Package File Locations

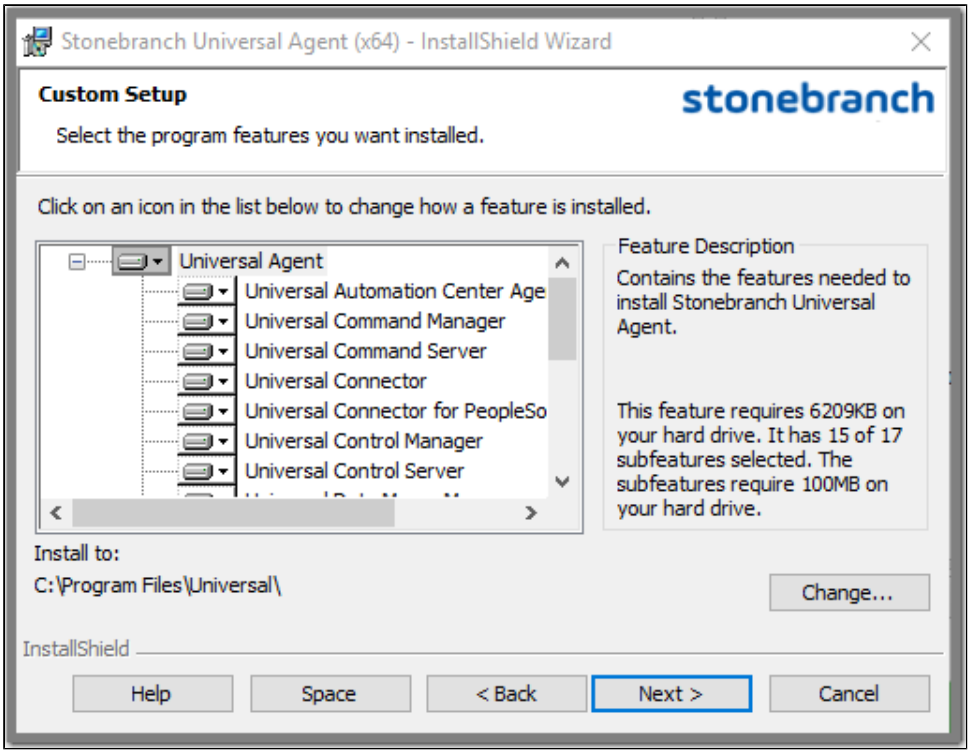
Installing a Universal Agent for Windows System Install via the Windows Installer Graphical Interface

To install a Universal Agent for Windows System Install via the Windows Installer graphical interface, perform the following steps:

Step 1	Download the desired Universal Agent for Windows System Install product distribution file to your work station: • **`sb-7.2.0.<level>-windows-x64.exe`**, the 64-bit Universal Agent for Windows System Install distribution file.
Step 2	Execute the distribution file from the command line to begin the installation process, which first determines whether a Windows Installer update is needed (see Windows Installer). The process then extracts and saves a Windows installer package file (.msi) to one of these locations.
Step 3	The installation starts after the files are extracted. It first will verify that your machine meets the minimum system requirements (see Installation Requirements and Summary - Windows). If the requirements are met, a Welcome dialog displays. 

Step 4

Click the **Next** button. A list of Universal Agent components included in the installation package then displays. It is from this list that you can select which components to install.



For a new installation, a drive icon displays next to each component in the list, indicating that the component will be installed.

Note

An **X** icon displays next to the Universal Controller Command Line Interface (CLI) component, indicating that, by default, it will not be installed. If you want to install the CLI, you must click the **X** icon and select the drive icon.

For an upgrade installation, either of the following icons displays next to an item:

- A drive icon indicates that the component is either:
  - New to the installation and will be installed.
  - Currently is installed and will be upgraded.
- An **X** icon indicates that the component is either:
  - Currently not installed (but previously was available).
  - Previously installed but removed.

A Stonebranch Tip

- If the installation detects an existing Universal Agent installation, currently installed components may be upgraded. (Currently, there is no way to specify that the state of a currently installed component should remain unchanged.)
- If a component is selected for installation, and the version of the installed component is less than the version of the component being installed, the installed component will be replaced by the component being installed.
- If a component is not selected for installation (that is, the **X** icon is selected), and it currently is installed, the new installation will remove that component.

Step 5

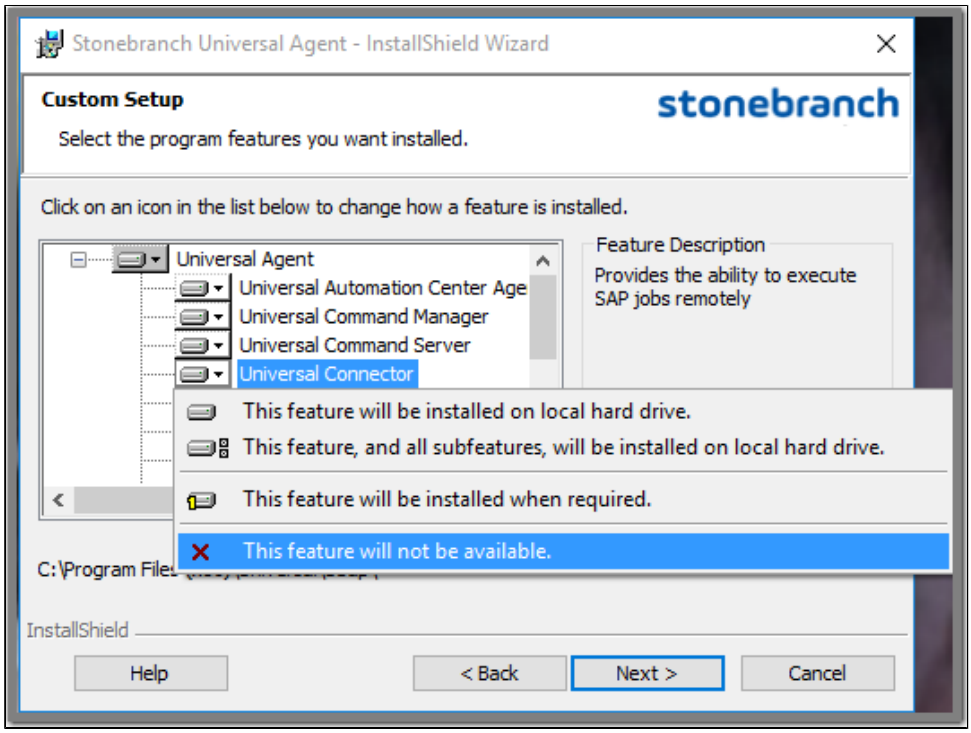
The previous figure indicates that all Universal Agent components will be installed in their respective directories under the **C:\ProgramFiles\Universal** directory.

- If you want to select a different location, click the **Change...** button.
- If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.

Step 6

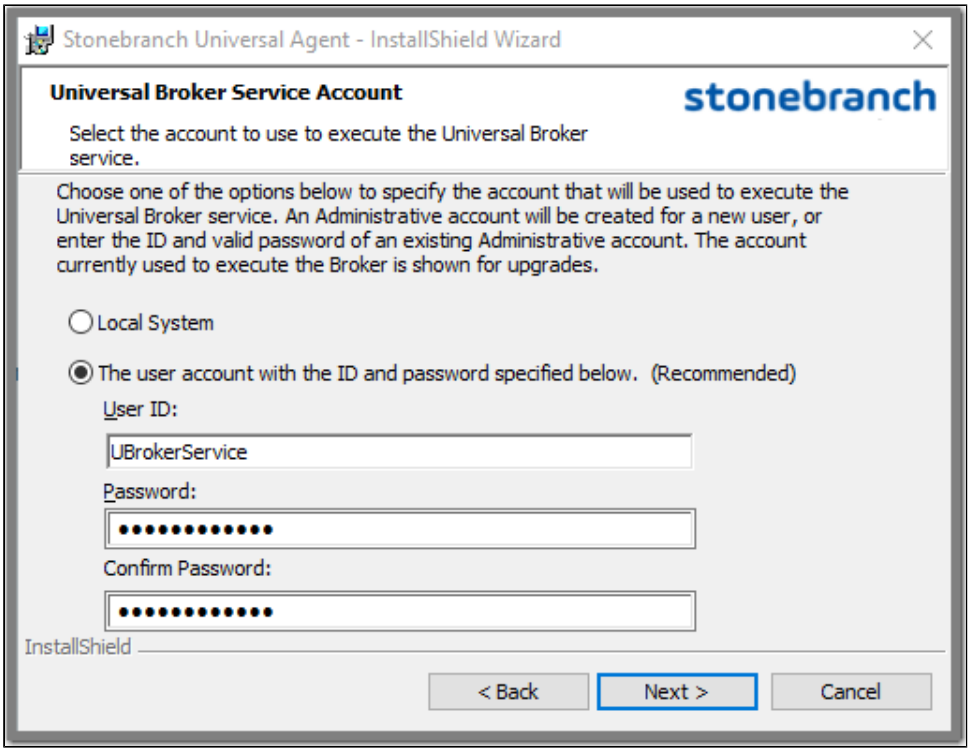
If you do not want to install a component:

1. Click the drive icon next to that component name.
2. From the drop-down list that displays, select the **X** icon to mark the component as one not to be installed.
   For example, the following figure indicates that Universal Connector has been selected to not be installed.



Step 7

When you have selected the components (and their installation destinations) that you want to install, click the **Next>** button. The Universal Broker Service Account dialog then displays.



Step 8

Select an account to use to execute the Universal Broker service:

- **Local System** (This is the default for upgrades from releases prior to 4.3.0.1.)
- An Administrative account capable of executing the Universal Broker service. (This is the default for new installs or upgrades where the Universal Broker currently is executing with an account other than **Local System**.)

  The user account ID defaults to **UBrokerService**, although any valid user name (up to 20 characters) can be specified.

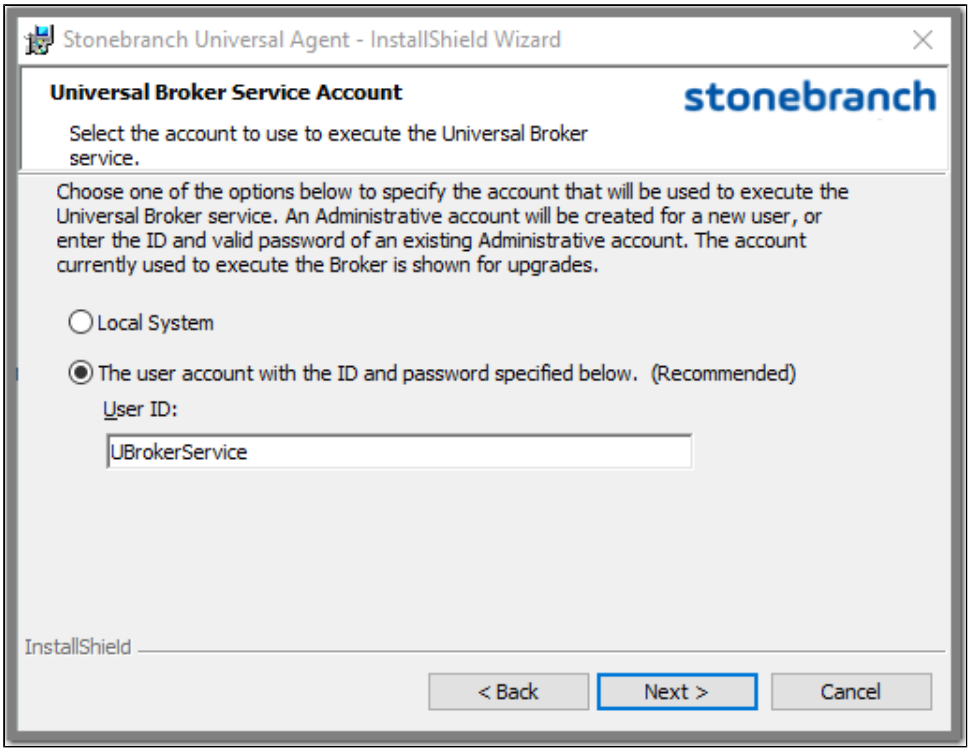
  If the account does not already exist, the install will create it with the privileges necessary to execute the Universal Broker service. See Starting Universal Broker for the privileges that an account needs to run the Universal Broker service.

  A default password provided for the account can be obtained by contacting Stonebranch support, although any password up to 256 characters can be specified.
  - If the user account does not exist, its password is set to the default value.
  - If the user account does exist, the installation process will validate the specified password. You must enter a valid password for the account to continue with the install.

The Broker service may also be run using a Windows domain account. The primary difference between using a local and domain account is that the domain account must already exist, and the installation will not configure it like it does a new local account. Required account privileges and file system access can be found in Service Security.

Note

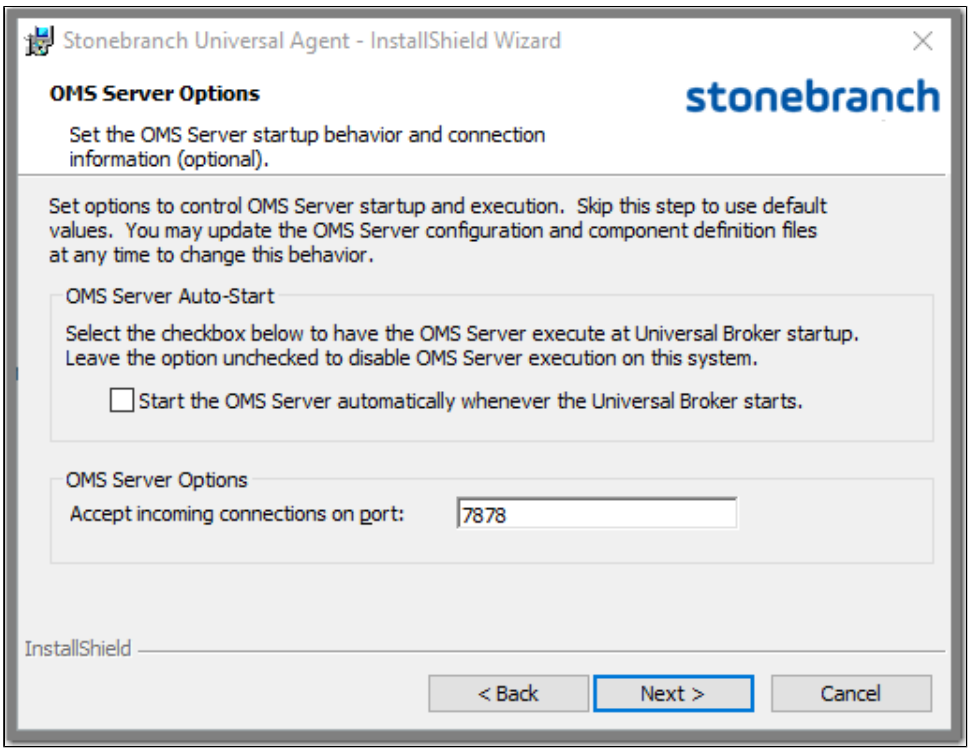
For upgrades, if the Universal Broker service is already configured to run with a user account, the install dialog shows only the account name and omits the password fields. The Broker service's configuration in the Windows Service Control Manager is retained across upgrades, making it unnecessary to re-enter account information during an upgrade.



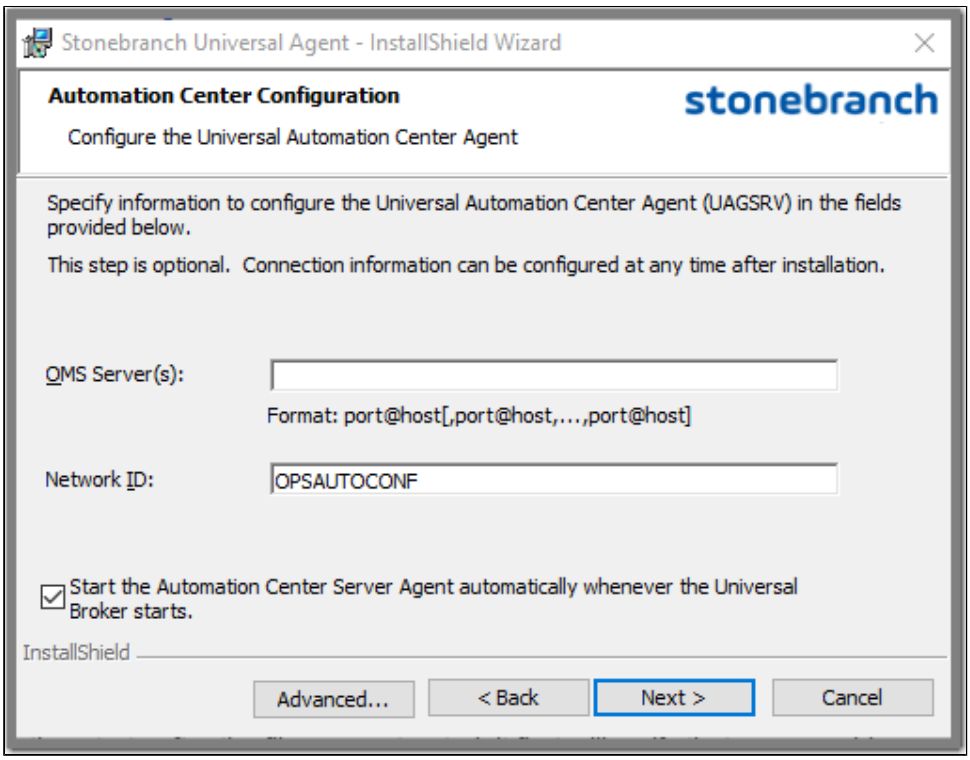
The install provides the RUNBROKERASUSER, BROKERUID, and BROKERPWD command line options to override the default behavior described above.

- Specifying RUNBROKERASUSER=0 causes the install to select Local System, regardless of how the Broker is currently configured to execute.
- Specifying RUNBROKERASUSER=1 causes the install to select the user account option, even if the Broker is currently configured to run as Local System.

When RUNBROKERASUSER=1, the BROKERUID and BROKERPWD command line options are provided to override the default or currently configured user ID and password. The BROKERUID option limits user IDs to 20 characters; the BROKERPWD option limits passwords to 14 characters. (See Windows Installer Command Line Parameters for more information).

Step 9	When you have selected an account, click the **Next>** button. The OMS Server Options dialog then displays.  Note This dialog displays only for new Universal Agent installs and upgrades to Universal Agent from earlier releases. It does not display if you selected, in Step 6, to not install Universal Message Service (OMS), or for upgrades from Universal Agent to a later version.
Step 10	Select whether or not you want the Universal Message Service (OMS) Server to start up when Universal Broker starts up and, optionally, change the OMS Server port connection (the default port, 7878, is pre-selected).

Step 11	Click the **Next>** button. The Automation Center Configuration dialog displays.



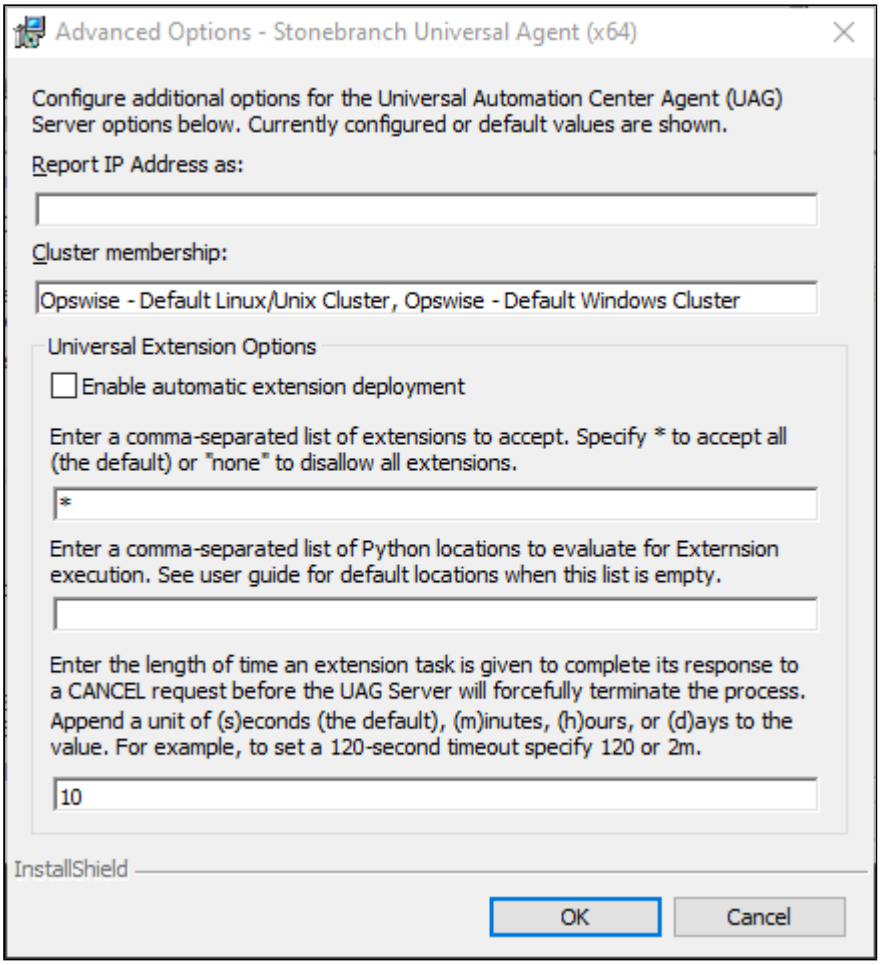
Step 12

Enter a list of OMS Servers to specify how the Agent will connect to Universal Controller.

If the **Start the Automation Center Server Agent automatically when the Universal Broker starts** field is enabled (the default), you must specify at least one OMS Server.

If you do not want the Agent to start automatically whenever the Universal Broker starts, deselect that field. You then do not have to specify any OMS Servers; this information can be configured after installation.

If you want to configure additional options for Universal Automation Center Agent (UAG), click the **Advanced** button. The Advanced Options dialog displays.

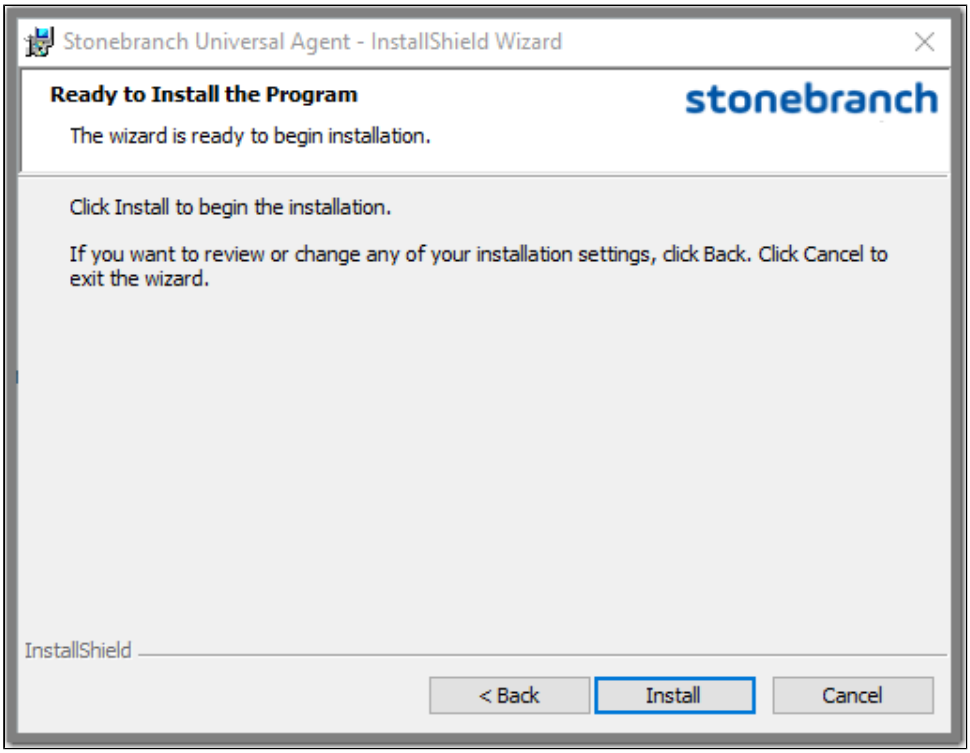


Configure the UAG options as desired and click the **OK** button.

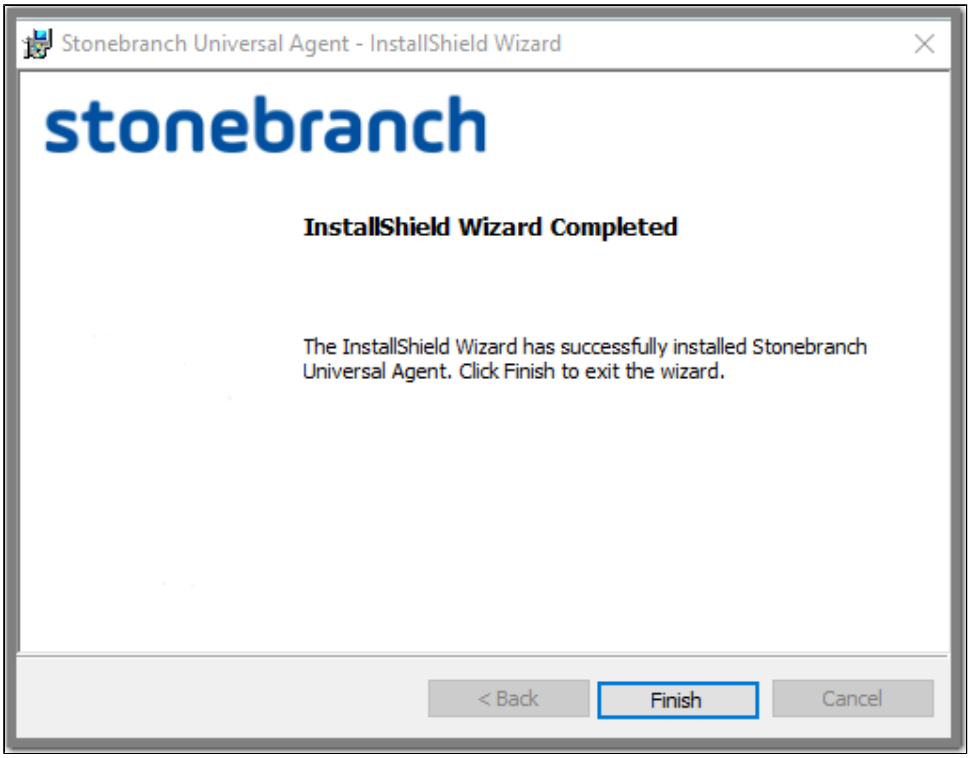
Step 13

Click the **Next>** button to continue the installation process. Depending on the components selected, the installation may prompt for additional values, such as working folders for Server components. Follow the directions provided with each dialog.

When the installation is ready to begin, the Ready to Install dialog displays.



Click the **Install** button to begin the installation or click the **<Back** button to return to change information on any of the previous dialogs. When the installation completes successfully, the Installation Complete dialog displays.



Step 14

Click the **Finish** button to exit Windows Installation.

Windows Installer Package File Locations

The Windows Installer Package file (**Ucmdx64.msi**) is extracted to the following location:

Universal Agent for Windows package (64-bit)	**<LocalAppData>\UniversalAgentx64\<packagecode>\Ucmdx64.msi**

In these paths:

- **<LocalAppData>** represents a particular user's local Application Data folder.
  For example: If the installation was performed by the built-in Administrator account, **<LocalAppData>** would expand by default to:
  - **C:\Users\Administrator\AppData\Local**.
- **<packagecode>** is a Universally Unique Identifier (UUID) in the format **{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}**, where 'X' is a hexadecimal character in the range 0 (zero) - F.
  For example: **{3B10285A-9602-4DC9-B0A5-4D701BEB5225}**.

Examples

In this example:

- Package level is 0 (zero).
- Install is executed by the built-in Administrator user.
- Package code for the 64-bit Universal Agent for Windows package is **{3B10285A-9602-4DC9-B0A5-4D701BEB5225}**.

Universal Agent for Windows Package (64-bit)

Distribution File	sb-7.2.0.0-windows-x64.exe
.msi File Name	Ucmdx68.msi
Default .msi File Location	**C:\Users\Administrator\AppData\Local\UniversalAgentx64\{3B10285A-9602-4DC9-B0A5-4D701BEB5225}**

Modifying a Universal Agent System Install via the Graphical Interface

- Modifying a Universal Agent for Windows System Install via the Windows Installer Graphical Interface
- Adding or Removing Universal Agent Components
- Repairing a Corrupted Universal Agent Installation
- Removing a Universal Agent Installation
    - Un-Installed Files

Modifying a Universal Agent for Windows System Install via the Windows Installer Graphical Interface

This page describes how to modify a Universal Agent for Windows System Install via the Windows Installer graphical interface.

After Universal Agent is installed, the Windows Installer can be run as many times as needed to modify the installation by:

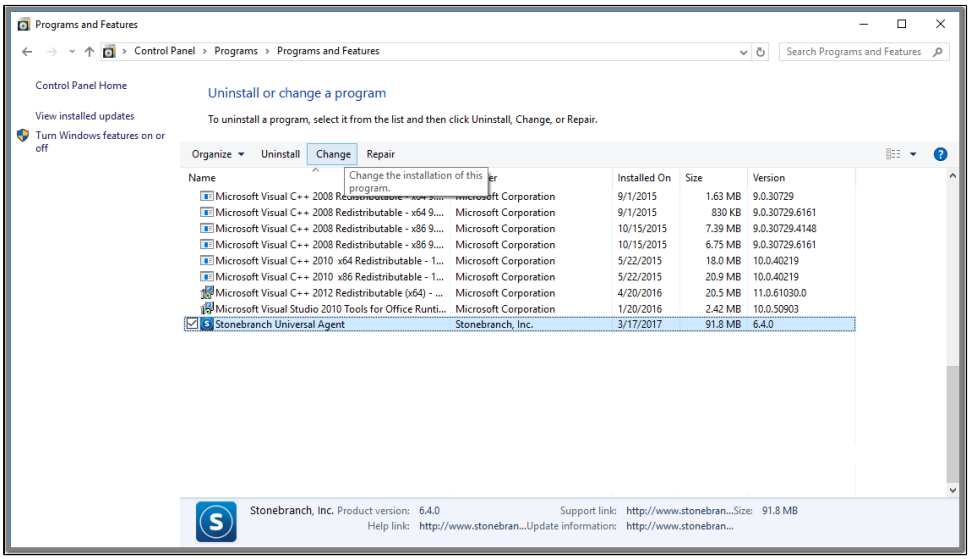
- Adding or Removing Universal Agent Components
- Repairing a Corrupted Universal Agent Installation
- Removing a Universal Agent Installation

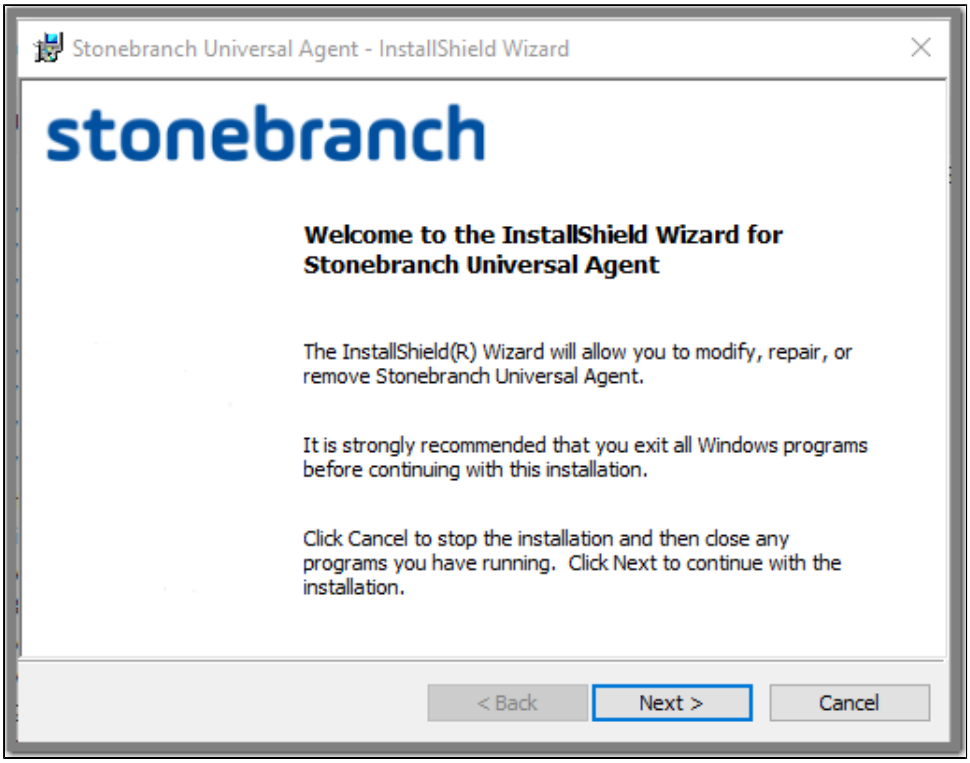
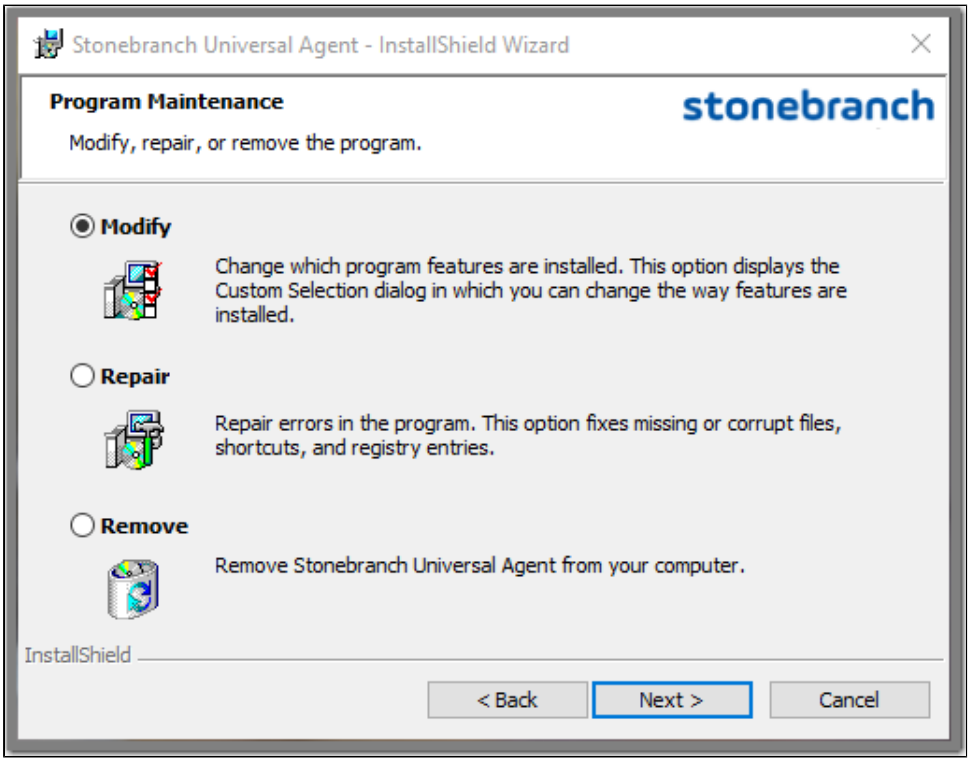
Adding or Removing Universal Agent Components

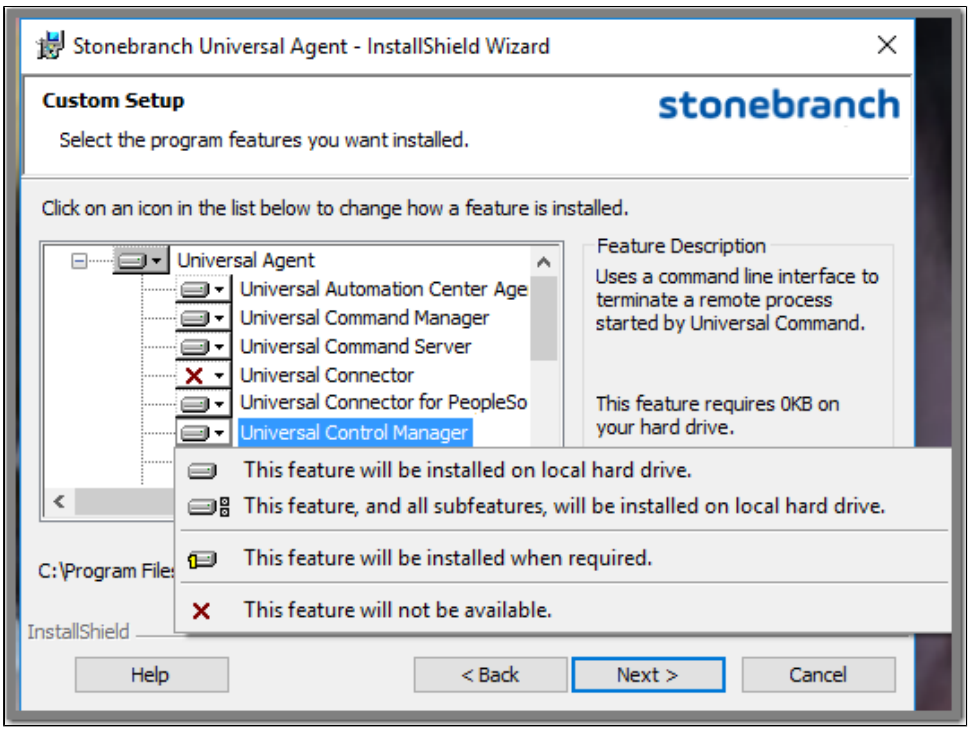
To add or remove components from a Universal Agent installation, perform the following steps:

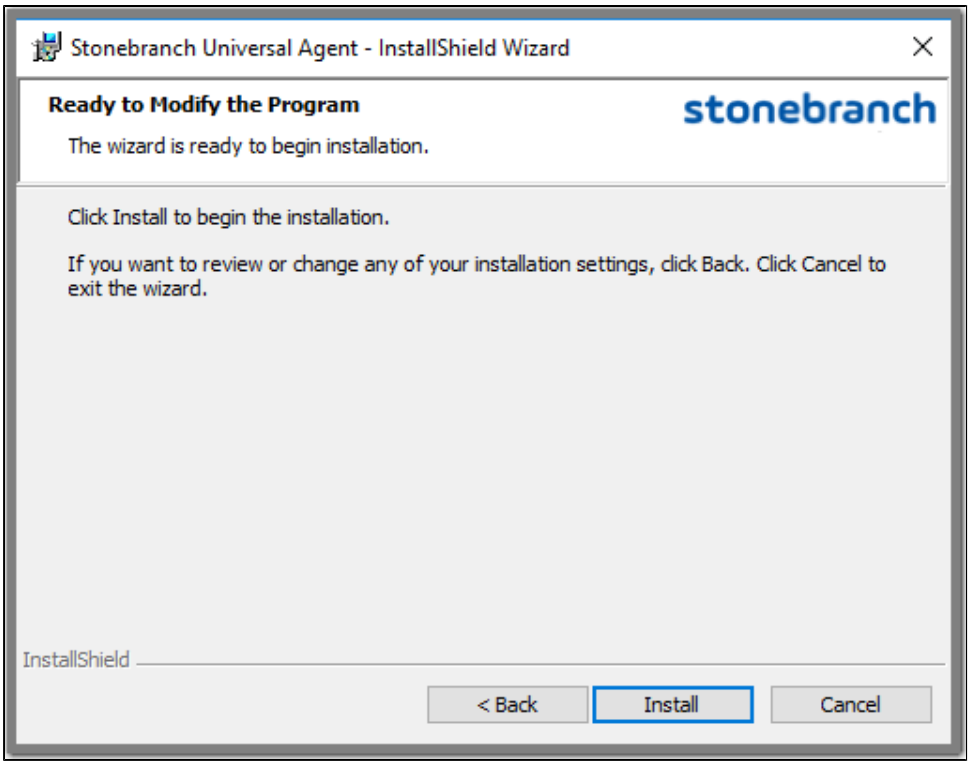
Step 1

On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.



Step 2	From the list of installed programs, select **Stonebranch Universal Agent** and click **Change** to start Windows Installer. The Welcome dialog displays. 
Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 

Step 4	Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog.  Currently installed components are identified by a drive icon. Uninstalled components are identified by an **X** icon.
Step 5	To remove a currently installed component: 1. Click the drive icon next to that component. 2. Select the X icon from the drop-down list to mark the component for removal.
Step 6	To add an uninstalled component: 1. Click the X icon next to that component. 2. Select the drive icon from the drop-down list to mark the component for installation.

Step 7	Click the Next> button to continue the installation process. Depending on the components selected, the installation may prompt for additional values, such as working folders for Server components. Follow the directions provided with each dialog. When the installation is ready to be modified, the Ready to Modify the Program dialog displays. 

Step 8	Click the **Install** button to modify the installation.
	
	When the modifications are complete, the following actions will be taken:
	
	• Components marked with a drive icon will:
	  • Remain installed if they already are installed.
	  • Be installed if they are not already installed.
	• Components marked with an **X** will:
	  • Remain uninstalled if they are not currently installed
	  • Be removed if they currently are installed.

Repairing a Corrupted Universal Agent Installation

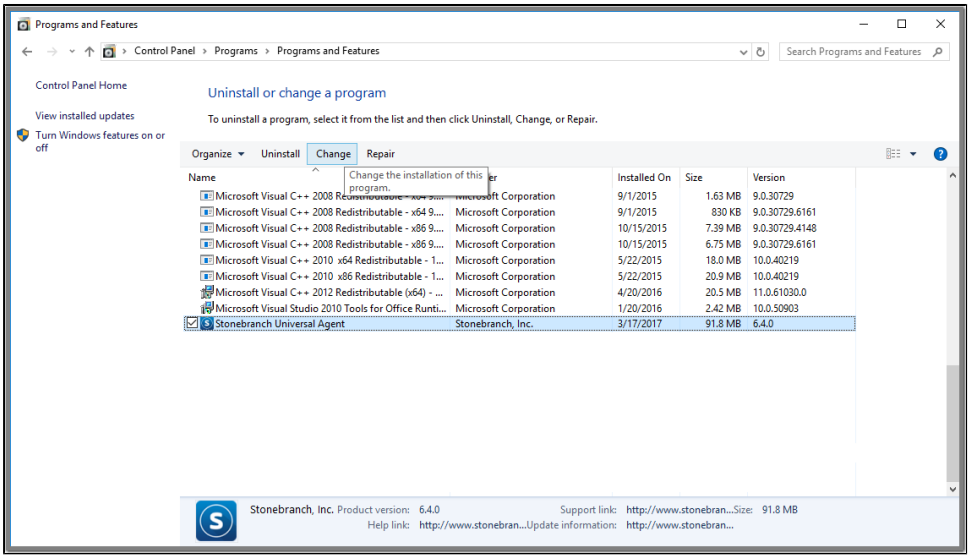
Windows Installer has the ability to recover accidentally deleted application files, configuration and component definition files, and registry entries required by Universal Agent. This repair feature will re-install the missing items, making a complete re-install unnecessary.

During a repair, any options stored in configuration and component definition files are preserved. If a component definition or configuration file was deleted, the installation will create a new configuration file with default values.

To repair an installation, perform the following steps:

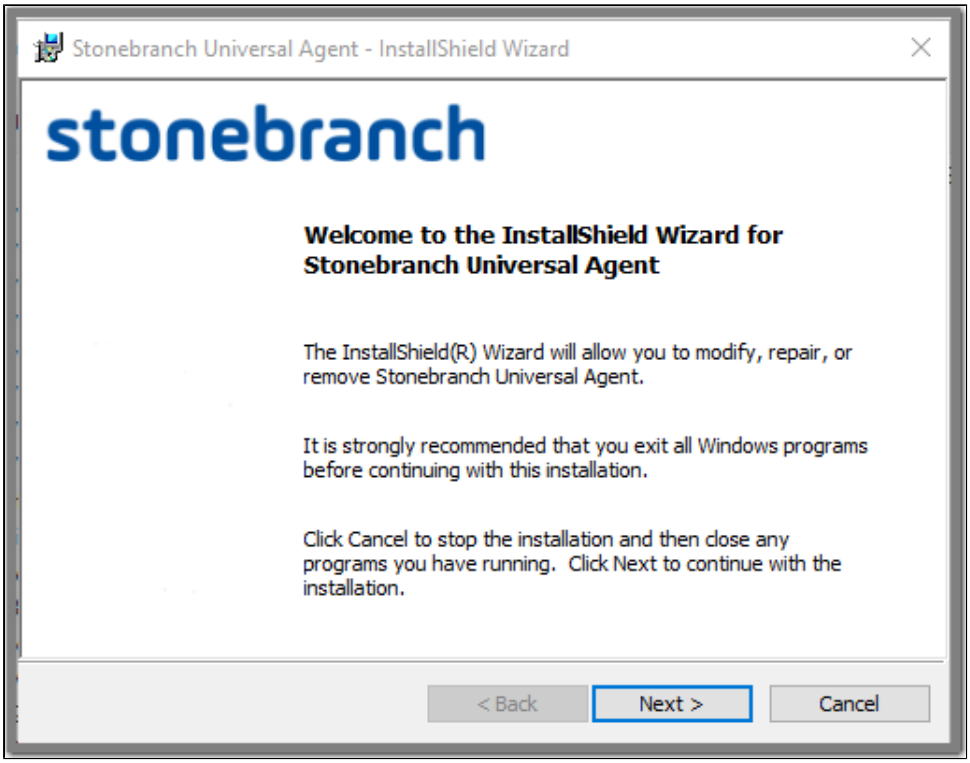
Step 1

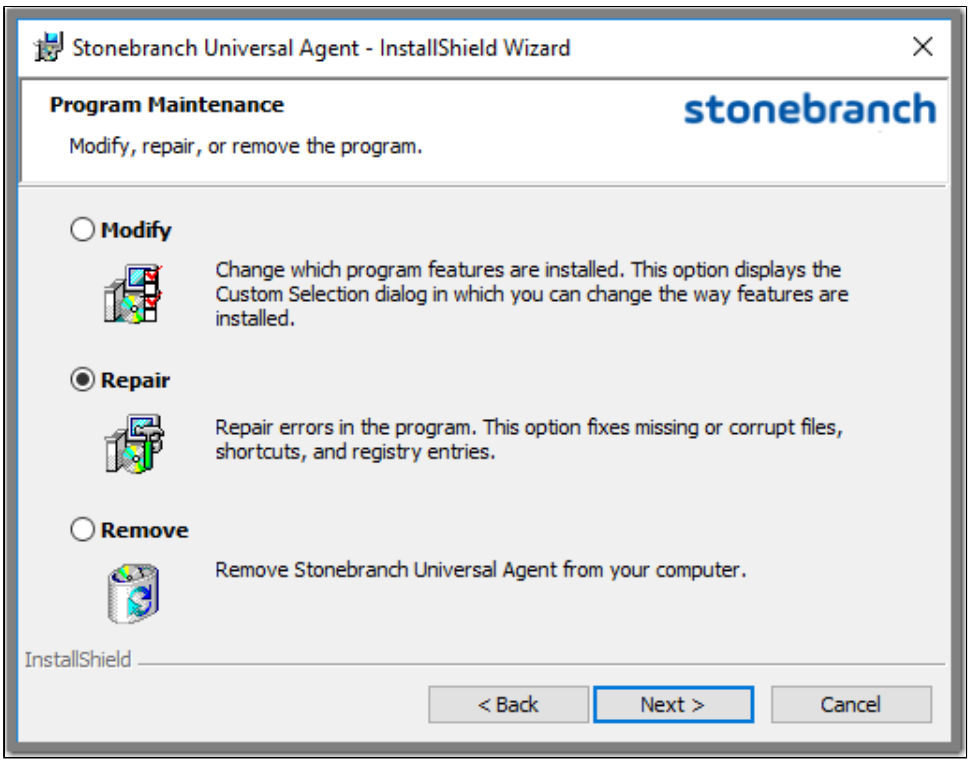
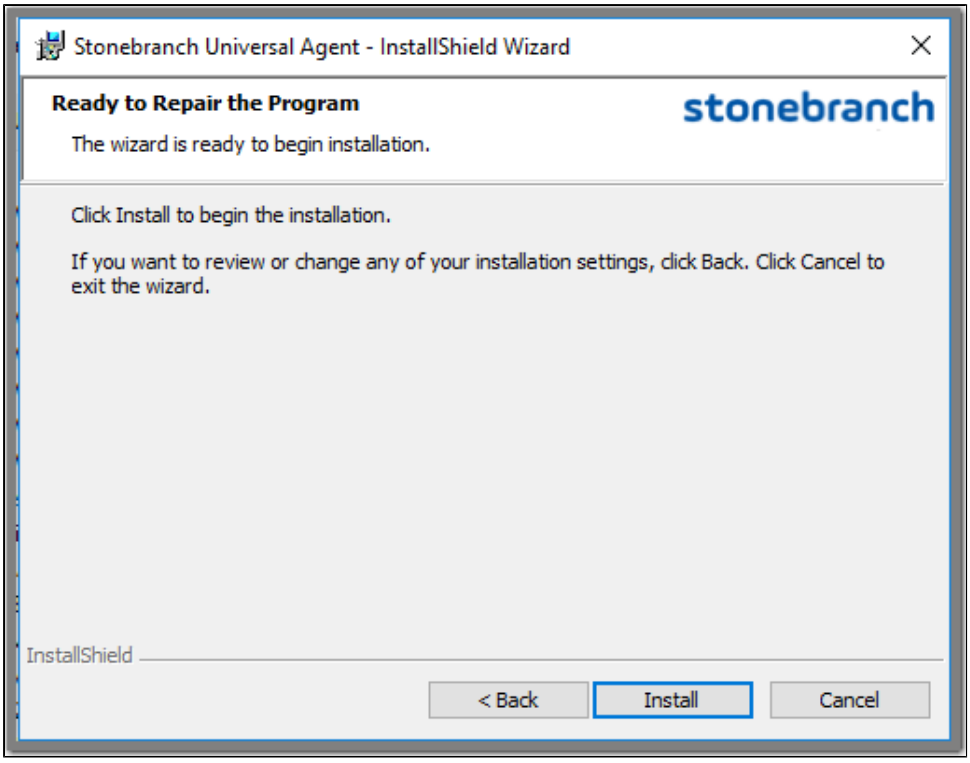
On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.



Step 2

From the list of installed programs, select **Stonebranch Universal Agent** and click **Change** to start Windows Installer. The Welcome dialog displays.



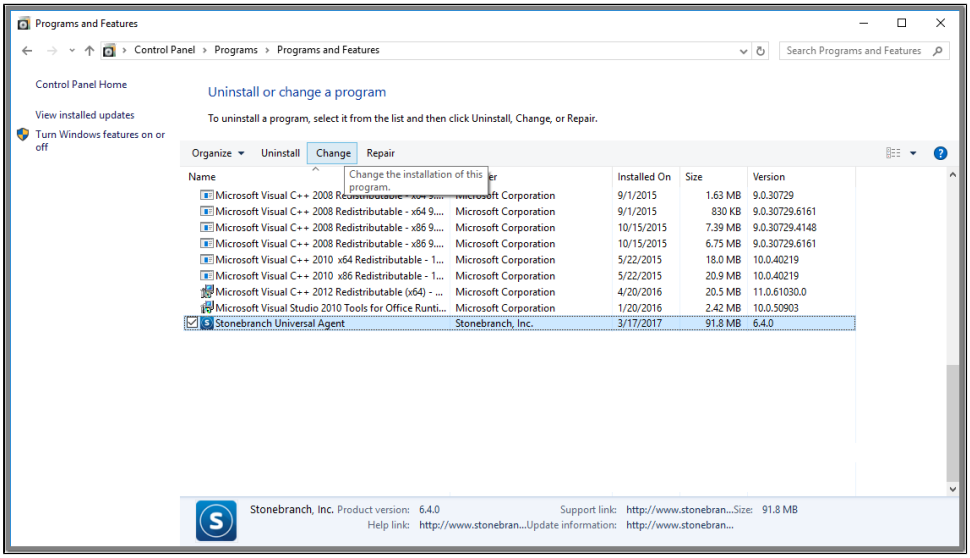
Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 
Step 4	Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair the Program dialog. 
Step 5	Click the **Install** button to repair the Stonebranch Universal Agent installation.

Removing a Universal Agent Installation

To uninstall a Universal Agent installation, perform the following steps:

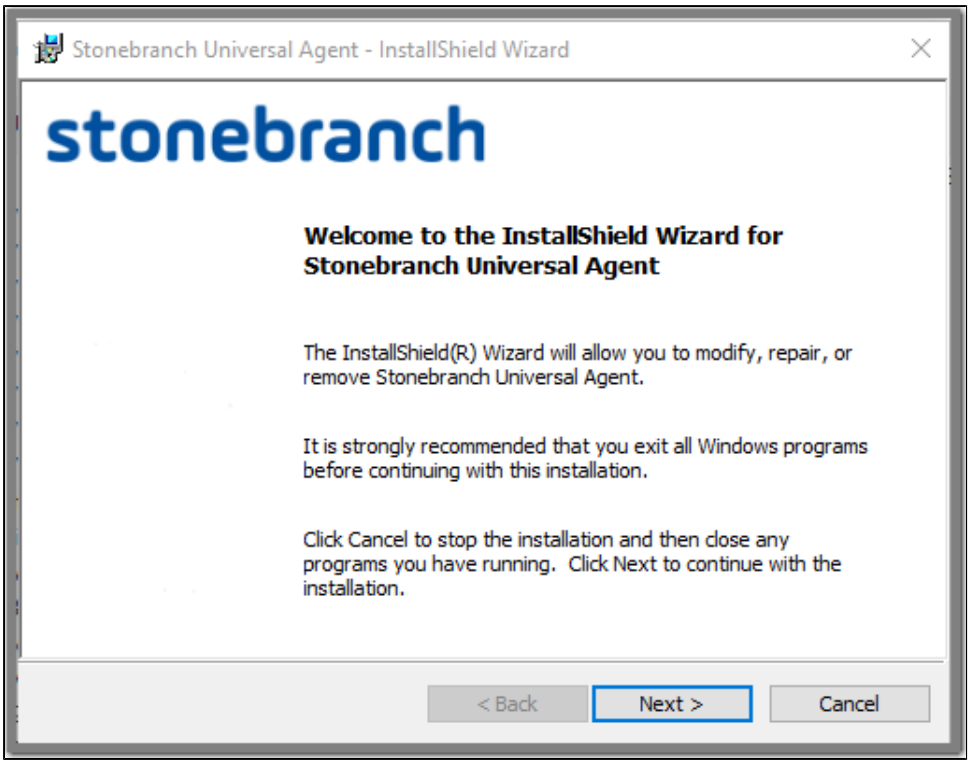
Step 1

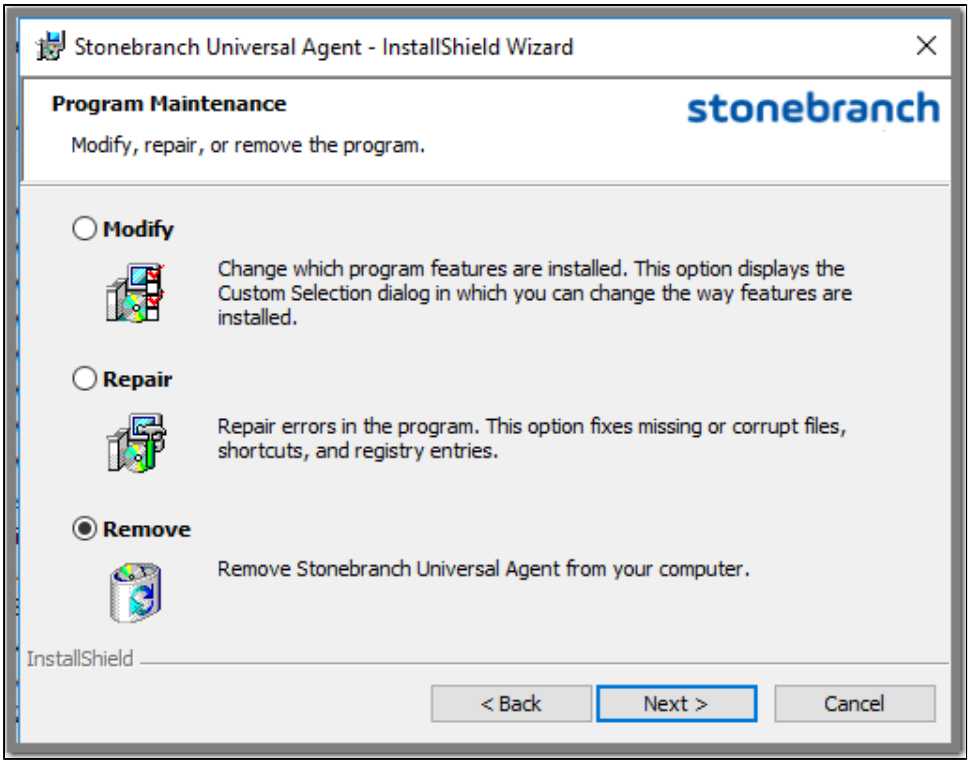
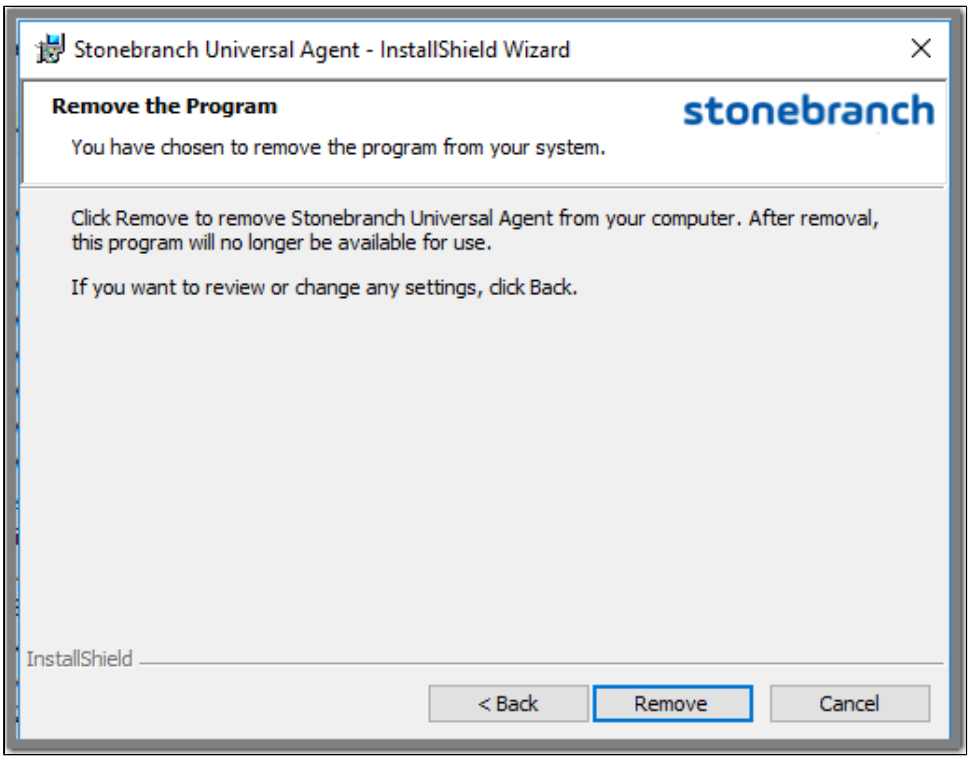
On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.



Step 2

From the list of installed programs, select **Stonebranch Universal Agent** and click **Change** to start Windows Installer. The Welcome dialog displays.



Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 
Step 4	Click the **Remove** radio button, and then the **Next>** button, to display the Remove the Program dialog. 
Step 5	Click the **Remove** button to remove the Stonebranch Universal Agent installation.

Un-Installed Files

The un-install process will remove only those files created during the installation. Some files stored under the **.\Universal** install directory by Universal Agent, such as trace files, files created under the Universal Command Server working directory, and spool files, may be left behind after the un- install. In this situation, those files and/or directories may simply be deleted.

Before deleting the entire **.\Universal** directory, make sure that no other Stonebranch, Inc. products are installed there. (See Universal Agent for Windows - System Installation File Inventory Lists for a list of files and directories installed with Universal Agent.)

In addition to those files and directories created by the Universal Agent installation, there may be some shared files (for example, codepage files) and Universal Agent components (for example, Universal Configuration Manager and Universal Encrypt) that may be left behind following an un-install. These components will be removed when the last Stonebranch Inc. product that uses them is un-installed.

Installing a Universal Agent System Install via the Command Line Interface

- Installing a Universal Agent for Windows System Install via the Windows Installer Command Line Interface
- Installing the Agent
  - Command Line Syntax
  - Command Line Switches
  - Command Line Parameters
  - Command Line Installation Examples
- Detecting the Completion of a Silent Install

Installing a Universal Agent for Windows System Install via the Windows Installer Command Line Interface

This page describes how to install a Universal Agent for Windows System Install using the Windows Installer command line interface.

A command line installation is useful in situations where:

- Several Universal Agent installations must be deployed across many different systems.
- It is not practical or convenient to perform the graphical interface installation.
- It is necessary to generate an installation log file.

Installing the Agent

To install a Universal Agent System Install using the Windows Installer command line interface, perform the following steps:

Step 1	Download the desired Universal Agent for Windows product distribution file to your work station: • **`sb-7.2.0.<level>-windows-x64.exe`**, the 64-bit Universal Agent for Windows distribution file (for supported 64-bit versions of Windows only).
Step 2	Execute the distribution file from the command line, and include all appropriate command line switches and parameters. The installation process determines whether a Windows Installer update is needed. The process then extracts and saves a Windows installer package file (.msi) to one of these locations. After all files (including the .msi) are extracted from the distribution file, the installation process verifies that your machine meets the minimum installation requirements. If the requirements are met, the installation begins.

Command Line Syntax

The following illustrates the command line syntax used to install the Agent:

```
sb-7.2.0.<level>-windows-<platform>.exe [/v"command line parameters"]  [/s] [/w] [/x]
```

In this syntax:

- **`<level>`** is the numeric package level.
- **`<platform>`** is **`x64`** for the 64-bit Universal Agent distribution file.

The command line switches (/v, /s, /w, and /x) are processed directly by the distribution file to control behavior of the Windows Setup application.

The command line parameters are passed to the Windows Installer (**msiexec**) to control the extracted Windows Installer Package file (**Ucmd.msi** or **Ucmd x64.msi**) behavior during the install process.

Command Line Switches

The following table describes the command line switches available for a command line installation:

/fom	Repairs a Universal Agent for Windows installation. **om** (after the **/f**) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the **om** options must be used. **/fom** cannot be used with the **/x** (uninstall) command line switch.
/s	Suppresses the initialization and extraction dialogs displayed before the product install Welcome dialog." If you are using the /q command line parameter, use this switch additionally for a completely silent install.
/v	Passes parameters to the Windows Installer (**msiexec**). The list of parameters must be enclosed in double (") quotation marks. See Command Line Parameters for available parameters.
/w	Instructs the Windows Setup application to wait until the installation completes. Use this switch when launching the installation from a script file. Without it, the Setup application may return immediately after launching Windows Installer.
/x	Uninstalls Universal Agent for Windows. **/x** cannot be used with the **/fom** (repair) command line switch.

Command Line Parameters

The following table describes the parameters that are available for a command line installation.

The parameters can be specified in any order, with the following exceptions:

- If the Silent install (/q) and/or Log file (/L) parameters are used, they can be specified in any order, but they must be specified ***after*** all other parameters.

These parameters are preceded by the /v command line switch and must be enclosed in double (") quotation marks.

Parameter	Description	Default
AC_AGENT_CLUSTERS=*clusters*	Sets the AGENT_CLUSTERS UAG configuration option to specify the Universal Controller-defined agent clusters to which this agent will belong.	(none)
AC_AGENT_IP=*IP address*	Sets the AGENT_IP UAG configuration option to have UAG Server register with Universal Controller using a specific IP address.	(none)
AC_ENABLE_SSL	*This option is deprecated starting with Universal Agent 7.1.0.0. UAG Server will always attempt to use SSL/TLS for OMS connections.*	n/a
AC_EXTENSION_ACCEPT_LIST	Sets the EXTENSION_ACCEPT_LIST UAG configuration option, which is a comma-separated list of Universal Extensions the Agent will accept. The default value for this option is "*", which will cause the Agent to accept all extensions. Specify a value of "none" to cause the Agent to disallow execution of all Extensions.	*
AC_EXTENSION_CANCEL_TIMEOUT=*value***[s|m|h|d]**	Sets the EXTENSION_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time that a Universal Extension task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (s)econds, (m)inutes, (h)ours, or (d)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	10
AC_EXTENSION_DEPLOY_ON_REGISTRATION	Sets the EXTENSION_DEPLOY_ON_REGISTRATION UAG configuration option that controls whether UAG Server will automatically receive Universal Extensions from Universal Controller at Agent startup.	NO

AC_EXTENSION_PYTHON_LIST	Sets the EXTENSION_PYTHON_LIST UAG configuration option, which is a comma-separated list of all Python interpreters that should be considered for Universal Extension execution. There are some default locations the Agent will check even if this option is not specified. See the EXTENSION_PYTHON_LIST option's description for more information.	(none)
AC_NETNAME=*value*	Specifies a value for a network ID that uniquely identifies the Agent on this system to the Universal Controller. The install uses this value to set the UAG NETNAME configuration option. The value specified in **AC_NETNAME** will override any currently configured options, including those imported from **`agent.props`**, if CONVERT_OPSAGENT is set to **yes**. If **AC_NETNAME** is omitted from the command line, and no other value specifies the Agent's network ID, the default is used.	OPSAUTOCONF
BROKERPWD=*BrokerAccountPassword*	Used by the install when RUNBROKERASUSER=**1** to specify the password for the account used to execute the Universal Broker service. The install also uses **BROKERPWD** to initialize the Password shown in the Universal Broker Service Account dialog. Valid values are any 256 characters or less. **BROKERPWD** is ignored if RUNBROKERASUSER=**0**. Note If the **BROKERPWD** option is set from the command line, and an installation log is generated (using the `/l` option to `msiexec`), the password value may be displayed in the log file, depending on the logging options used. Specifically, the c, p, and v flags will show the value of the **BROKERPWD** command line option. We recommend using `/liarewum` to set installation log file options to collect as much information as possible without exposing any sensitive information. If the **BROKERPWD** option is **not** used from the command line, all logging options may be turned on (using `/l*v`) without exposing any sensitive information.	(none)
BROKERUID=*BrokerAccountId*	Used by the install when RUNBROKERASUSER=**1** to override the default or currently configured user ID associated with the account used to execute the Universal Broker service. The install also uses **BROKERUID** to initialize the User ID shown in the Universal Broker Service Account dialog. Valid values are any 20 characters or less. For domain accounts, a domain name of up to 256 characters is accepted, but the user ID is still limited to 20 characters. **BROKERUID** is ignored if RUNBROKERASUSER=0.	(none)
CONVERT_OPSAGENT={**yes**|**no**}	Specification for whether or not to convert an existing Opswise Automation Center Agent (1.5, 1.6, or 1.7) to Agent 7.2.x. The installation process will invoke a script, **`opsmerge.vbs`**, which stops the Agent and converts the configuration options stored in its **`agent.props`** file to corresponding configuration options in the Universal Automation Center Agent (UAG) **`uags.conf`** file. The script also performs several other tasks needed for the conversion. Note You must include this parameter when upgrading an Opswise Automation Center Agent (1.5, 1.6, or 1.7) to Agent 7.2.x.	**no**
INSTALLDIR=*installdir*	Sets the root installation directory to *installdir*. Each component will be installed under this directory. **INSTALLDIR** is required only if you want to install Universal Agent for Windows under a directory different from the one specified by the **PROGRAMFILES** environment variable (typically **`C:\Program Files\Universal`**). If the directory contains spaces, you must use double (**"**) quotation marks around the path name.	(none)
KEYSTORE={**yes**|**no**}	Specification for whether or not to create encryption keys during installation and set up the local Universal Broker as a keystore owner.	
OMS={**yes**|**no**}	Specification for whether or not to install the Universal Message Service (OMS) during new installs, upgrades, or maintenance. • If **yes** is specified, OMS will be installed. • If **no** is specified, OMS will not be installed. If OMS already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **OMS** is not required unless you want to change the current install state. **OMS** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Message Service** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**

OMS_AUTOSTART={**yes**|**no**}	Specification for setting the start-up behavior of the Universal Message Service (OMS) Server component. • If **yes** is specified, the AUTOMATICALLY_START OMS component definition option is set to **yes**, which instructs Universal Broker to start the OMS Server automatically when it is started. • If **no** (the default) is specified, or if this option is not included on the command line, the AUTOMATICALLY_START and RESTART OMS component definition options are set to **no**. Universal Broker will not start the OMS Server automatically when it is started. This allows the OMS Server to be started manually any time after the Broker is started. This OMS_AUTOSTART value is used to initialize the state of a check box on the OMS Server Start-up Option dialog, which displays for new installs and upgrades from Universal Agent releases earlier than 5.2.0. Changing the state of this check box will override the value specified from the command line.	**no**
OMS_SERVERS=*port@ipaddr*[,_port**n**@ipaddr**n**_,...]	A list of one or more locations where an OMS Server resides.	(none)
OPSCLI={**yes**|**no**}	Specification (yes or no) for whether or not to install the Universal Controller Command Line Interface (CLI) during new installs, upgrades, or maintenance. • If **yes** is specified, the CLI will be installed. • If **no** is specified, the CLI will not be installed. If the CLI already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **OPSCLI** is not required unless you want to change the current install state. **OPSCLI** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Controller Command Line Interface** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**no**
PYTHON={**yes**|**no**}	Specification for whether or not the Python 3.7 Distribution for Universal Agent is installed.	no
RUNBROKERASUSER={**0**|**1**}	Controls whether the Universal Broker service executes as an Administrative user account or as **Local System**. • If **RUNBROKERASUSER**=**1**, the install will perform the following steps: 1. If the Universal Broker service is not installed or currently is configured to run as **Local System**, the install will set the service's start-up account to **UBrokerService** or the user ID specified by the BROKERUID command line option. 2. If the Universal Broker service is installed and already is configured to run as a user account, the Broker's start-up account will not be changed (allows customized user accounts to be preserved during upgrades). • If **RUNBROKERASUSER**=**0**, the install will perform the following steps: 1. If the Universal Broker service is not installed or is currently configured to run as an account other than **Local System**, the install will set the service's start-up account to **Local System**. Note The Universal Broker service's properties will be set to allow Universal Broker to interact with the desktop. This is supported only when Universal Broker runs as Local System. 2. If the Universal Broker service is installed and is currently configured to run as **LocalSystem**, the Universal Broker service's properties will not be changed.	**1**: new installs **0**: upgrades earlier than 4.3.0
UAG_AUTOSTART={**yes**|**no**}	Sets the AUTOMATICALLY_START UAG server component definition option to control whether Universal Broker automatically starts the UAG Server.	**yes**
UAGSRV={**yes**|**no**}	Specification for whether or not to install Universal Automation Center Agent (UAG) during new installs, upgrades, or maintenance. • If **yes** is specified, UAG will be installed. • If **no** is specified, UAG will not be installed. If UAG already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UAGSRV** is not required unless you want to change the current install state. **UAGSRV** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Automation Center Agent** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**

UCMDMGR={**yes**|**no**}	Specification for whether or not to install the Universal Command (UCMD) Manager during new installs, upgrades, or maintenance. • If **yes** is specified, the UCMD Manager will be installed. • If **no** is specified, the UCMD Manager will not be installed. If the UCMD Manager already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UCMDMGR** is not required unless you want to change the current install state. **UCMDMGR** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Command Manager** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UCMDSRV={**yes**|**no**}	Specification for whether or not to install the Universal Command (UCMD) Server during new installs, upgrades, or maintenance. • If **yes** is specified, the UCMD Server will be installed. • If **no** is specified, the UCMD Server will not be installed. If the UCMD Server already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UCMDRSV** is not required unless you want to change the current install state. **UCMDSRV** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Command Server** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UCMDSRV WORKFOLDER=*WorkFolderPath*	Sets the Universal Command Server working folder. This value is used only if the Universal Command Server is being installed. If the folder contains spaces, you must use double (**"**) quotation marks around the path name.	**INSTALLDIR\UcmdHome**
UCTLMGR={**yes**|**no**}	Specification (yes or no) for whether or not to install the Universal Control (UCTL) Manager during new installs, upgrades, or maintenance. • If **yes** is specified, the UCTL Manager will be installed. • If **no** is specified, the UCTL Manager will not be installed. If the UCTL Manager already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UCTLMGR** is not required unless you want to change the current install state. **UCTLMGR** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Control Manager** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UCTLSRV={**yes**|**no**}	Specification for whether or not to install the Universal Control (UCTL) Server during new installs, upgrades, or maintenance. • If **yes** is specified, the UCTL Server will be installed. • If **no** is specified, the UCTL Server will not be installed. If the UCTL Server already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UCTLRSV** is not required unless you want to change the current install state. **UCTLSRV** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Control Server** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UCTLSRVWORKFOLDER=*WorkFolderPath*	Sets the Universal Control Server working folder. This value is used only if the Universal Control Server is being installed. If the folder contains spaces, you must use double (**"**) quotation marks around the path name.	**INSTALLDIR\UcmdHome**
UDMMGR={**yes**|**no**}	Specification for whether or not to install the Universal Data Mover (UDM) Manager during new installs, upgrades, or maintenance. • If **yes** is specified, the UDM Manager will be installed. • If **no** is specified, the UDM Manager will not be installed. If the UDM Manager already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UDMMGR** is not required unless you want to change the current install state. **UDMMGR** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Data Mover Manager** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**

UDMSRV={**yes**|**no**}	Specification (yes or no) for whether or not to install the Universal Data Mover (UDM) Server during new installs, upgrades, or maintenance. • If **yes** is specified, the UDM Server will be installed. • If **no** is specified, the UDM Server will not be installed. If the UDM Server already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UDMRSV** is not required unless you want to change the current install state. **UDMSRV** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Data Mover Server** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UDMSRVWORKFOLDER=*WorkFolderPath*	Sets the Universal Data Mover Server working folder. This value is used only if the Universal Data Mover Server is being installed. If the folder contains spaces, you must use double (**"**) quotation marks around the path name.	**INSTALLDIR\UdmHome**
UELD={**yes**|**no**}	Specification for whether or not to install Universal Event Log Dump (UELD) during new installs, upgrades, or maintenance. • If **yes** is specified, UELD will be installed. • If **no** is specified, UELD will not be installed. If UELD already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UELD** is not required unless you want to change the current install state. **UELD** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Event Log Dump** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UEMMGR={**yes**|**no**}	Specification for whether or not to install the Universal Event Monitor (UEM) Manager during new installs, upgrades, or maintenance. • If **yes** is specified, the UEM Manager will be installed. • If **no** is specified, the UEM Manager will not be installed. If the UEM Manager already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UEMMGR** is not required unless you want to change the current install state. **UEMMGR** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Event Monitor Manager** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UEMSRV={**yes**|**no**}	Specification for whether or not to install the Universal Event Monitor (UEM) Server during new installs, upgrades, or maintenance. • If **yes** is specified, the UEM Server will be installed. • If **no** is specified, the UEM Server will not be installed. If the UEM Server already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UEMRSV** is not required unless you want to change the current install state. **UEMSRV** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Event Monitor Server** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
UEMSRVWORKFOLDER=*WorkFolderPath*	Sets the Universal Event Monitor Server working folder. This value is used only if the Universal Event Monitor Server is being installed. If the folder contains spaces, you must use double (**"**) quotation marks around the path name.	**INSTALLDIR\UemHome**
UPPS={**yes**|**no**}	Specification for whether or not to install the Universal Connector for PeopleSoft (UPPS) during new installs, upgrades, or maintenance. • If **yes** is specified, UPPS will be installed. • If **no** is specified, UPPS will not be installed. If UPPS already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UPPS** is not required unless you want to change the current install state. **UPPS** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Connector for PeopleSoft** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**

UQUERY={**yes**|**no**}	Specification for whether or not to install the Universal Query (UQUERY) during new installs, upgrades, or maintenance. • If **yes** is specified, UQUERY will be installed. • If **no** is specified, UQUERY will not be installed. If UQUERY already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UQUERY** is not required unless you want to change the current install state. **UQUERY** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Query** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
USAP={**yes**|**no**}	Specification for whether or not to install the Universal Connector (USAP) during new installs, upgrades, or maintenance. • If **yes** is specified, USAP will be installed. • If **no** is specified, USAP will not be installed. If USAP already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **USAP** is not required unless you want to change the current install state. **USAP** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Connector** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
USPOOL={**yes**|**no**}	Specification (yes or no) for whether or not to install the Universal Spool Utilities, Universal Spool List (USLIST) and Universal Spool Remove (USLRM), during new installs, upgrades, or maintenance. • If **yes** is specified, the Universal Spool Utilities will be installed. • If **no** is specified, the Universal Spool Utilities will not be installed. If the Universal Spool Utilities already are present on the system (via a previous installation), they will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **USPOOL** is not required unless you want to change the current install state. **USPOOL** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **Universal Spool Utilities** on the Custom Setup dialog when installing Universal Agent via the graphical interface.	**yes**
/q{**n**|**b**|**r**}	Suppresses the product installation dialogs. Use this parameter in addition to the #/s command line switch for a completely silent install. The following flags are available for **/q**: • **n** - Creates no user interface. • **b** - Creates a basic user interface, which displays a simple dialog box to report progress and any errors. • **r** - Creates a reduced user interface, which displays progress and errors in customized windows. See Command Line Switches, Command Line Installation Examples, and Detecting the Completion of a Silent Install for additional information regarding silent installs.	n/a
/L[***v**]	Instructs the installation process to create an installation log file named <**logfilepath**> (full path name). If <**logfilepath**> contains spaces, you must enclose it with double (**"**) quotation marks. ***v** are flags used to specify the level of detail (**verbose**) contained in the log file. To reduce the amount of output generated, ***v** can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.	n/a

Command Line Installation Examples

The following examples illustrate different ways that Universal Agent for Windows can be installed from the command line.

Graphical User Interface Install, All Components

To install all Universal Agent for Windows components via the graphical interface using the 32-bit distribution file, issue the following command:

```
sb-7.2.0.0-windows-x64.exe
```

Graphical User Interface Install, with Log File

```
sb-7.2.0.0-windows-x64.exe /v"/l*v C:\Temp\install.log"
```

Silent Install, All Components, with Log File

To silently install all 64-bit Universal Agent for Windows components using the 64-bit distribution file and write a log file to **`C:\Temp\windows install.log`**, issue the following command:

```
sb-7.2.0.0-windows-x64.exe /s /v"/qn /l*v C:\Temp\install.log"
```

Silent Install, Different Installation Directory

```
sb-7.2.0.0-windows-x64.exe /s /v"/qn INSTALLDIR=D:\Universal"
```

Silent Install, Quotation Marks for Directory Path with a Space

```
sb-7.2.0.0-windows-x64.exe /s /v"/qn INSTALLDIR=\"C:\Program Files\Stonebranch\""
```

Silent Installation Removal

```
sb-7.2.0.0-windows-x64.exe /x /s /v"/qn"
```

Install Partial Components to a Different Installation Directory

To install only the Universal Broker (which always is installed), Universal Data Mover Manager, Universal Event Monitor Manager, Universal Control Manager, and Universal Query components under **`D:\Universal`** (that is, a directory other than the one specified by the environment variable PROGRAMFILES) using the 32-bit Universal Agent distribution file, issue the following command:

```
sb-7.2.0.0-windows-x64.exe /v"INSTALLDIR=D:\Universal OMS=no UAGSRV=no UCMDMGR=no UCMDSRV=no UCTLSRV=no UELD=no
USPOOL=no UDMSRV=no UEMSRV=no"
```

Install All Components, Set OMS Server Location

To install all Universal Agent components using the 32-bit Universal Agent distribution file, and to set the Universal Automation Center Agent (UAG) OMS_SERVERS configuration option, issue the following command:

```
sb-7.2.0.0-windows-x64.exe /v"OMS_SERVERS=7878@opshost"
```

Install All Components, Set Universal Broker Service

To install all Universal Agent components using the 64-bit Universal Agent distribution file, and to set the Universal Broker service to execute with an administrative account with the specified user ID and password, issue the following command (for example):

```
sb-7.2.0.0-windows-x64.exe /v"RUNBROKERASUSER=1 BROKERUID=ubradmin BROKERPWD=guessme"
```

Note

If the option to generate an installation log file is used alongside the BROKERPWD option, the password value may be displayed in the log file, depending on the logging options used. Specifically, the c, p, and v flags will cause the value of the **BROKERPWD** option to appear in the log file.

To generate a log file when specifying the **BROKERPWD** option, we recommend using `/liarewum` to set to collect as much information as possible without exposing any sensitive information.

For example:

```
sb-7.2.0.0-windows-x64.exe /v"/liarewum setup.log RUNBROKERASUSER=1 BROKERUID=ubradmin BROKERPWD=guessme"
```

If the **BROKERPWD** option is **not** used from the command line, all logging options may be turned on (using `/l*v`) without exposing any sensitive information.

Detecting the Completion of a Silent Install

If /s and /q are used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided that indicates when the Windows Installer process (install, uninstall, or repair) finishes.

One method that can be used to detect when the installation process ends is to execute it using the system's **start** command. Using available command line switches, **start** can be used to initiate the installation process and then wait for it to finish. When **start** returns control to its calling process (for example, the command prompt), the process will have ended.

For example, from the command prompt, issue the following command to perform a silent 64-bit Universal Agent installation using the 64-bit distribution file and wait for it to finish:

```
start /b /wait sb-7.2.0.0-windows-x64.exe /w /s /v"/qn"
```

- The **/b** switch prevents the **start** command from opening a new window.
- The **/wait** parameter causes the **start** command to start the installation and then wait for it to finish.

This also is the recommended way to execute the installation from within a batch file.

For more information on the **start** command, go to the Windows command prompt and enter: **start /?**.

Modifying a Universal Agent System Install via the Command Line Interface

- Modifying a Universal Agent System Install via the Windows Installer Command Line Interface
- Adding or Removing Universal Agent Components
- Repairing a Corrupted Universal Agent Installation
  - Silent Repair, Distribution File Execution, with Log File
  - Silent Repair, 64-bit Installation, Windows Installer Execution, with Log File
- Removing a Universal Agent Installation
  - Interactive Removal, Distribution File Execution
  - Silent Removal, 64-bit Installation, Windows Installer Execution

Modifying a Universal Agent System Install via the Windows Installer Command Line Interface

This page describes how to modify a Universal Agent installation via the Windows Installer command line interface.

After Universal Agent is installed, Windows Installer can be run as many times as needed to modify the installation by:

- Adding or Removing Universal Agent Components
- Repairing a Corrupted Universal Agent Installation
- Removing a Universal Agent Installation

(For a description of the parameters used in these procedures, see Windows Installer Command Line Parameters.)

Adding or Removing Universal Agent Components

Currently, it only is possible to add or remove individual Universal Agent components using the Windows Installer graphical interface (see Modifying a Universal Agent System Install via the Graphical Interface).

Repairing a Corrupted Universal Agent Installation

To recover accidentally deleted files or registry entries required by Universal Agent using the Windows Installer command line interface, use the **/f** switch together with the **om** parameters.

These are the same repair options set internally by the graphical interface installation. They cause Windows Installer to reinstall files that either are missing or older than the version contained in the Universal Agent distribution file.

Silent Repair, Distribution File Execution, with Log File

To repair a Universal Agent installation from the command line using the original distribution file, issue the following command:

```
sb-7.0.0.<level>-windows-<platform>.exe /v"/fom /q /L*v c:\temp\repair.log"
```

In this command, **<level>** is the numeric package level and **<platform>** is x64 (64-bit distribution file).

Silent Repair, 64-bit Installation, Windows Installer Execution, with Log File

To repair a 64-bit Universal Agent installation using the **.msi** file that was extracted from the distribution file during the initial install, issue the following command:

```
msiexec.exe /fom SetupPath\Ucmdx64.msi /q /L*v c:\temp\repair.log
```

In the last two examples, **SetupPath** refers to the location in which the extracted **.msi** file resides. The exact path varies, depending on the user account that performed the install and the Windows version.

For information on how the installation determines **SetupPath**, see Step 2 of the Installing via Command Line procedures.

Note

All of these commands also use the optional **/L*v** parameter to generate a log file named **C:\temp\repair.log**. If you want to generate a log file, you can substitute the path and file name for one of your choosing. If no log file is desired, simply omit the **/L*v** option and file name.

Removing a Universal Agent Installation

Interactive Removal, Distribution File Execution

To uninstall a Universal Agent installation from the command line using the original distribution file, issue the following command:

```
sb-7.0.0.<level>-windows-<platform>.exe /x
```

In this command, **<level>** is the numeric package level and **<platform>** is x64 for the 64-bit distribution file.

Silent Removal, 64-bit Installation, Windows Installer Execution

To uninstall a 64-bit Universal Agent installation using the **.msi** file that was extracted from the distribution file during the initial install, issue the following command:

```
msiexec.exe /x SetupPath\Ucmdx64.msi /q
```

In the last two examples, **SetupPath** refers to the location in which the extracted **.msi** file resides. The exact path varies depending on the user account that performed the install and the Windows version.

For information on how the installation determines **SetupPath**, see Step 2 of the Installing via Command Line procedures.

Universal Agent for Windows - System Installation File Inventory Lists

- Universal Agent for Windows - File Inventory Lists
- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Connector for PeopleSoft
- Universal Connector for SAP
- Universal Control Manager
- Universal Control Server
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Configuration Manager
- Universal Copy
- Universal Encrypt
- Universal Event Log Dump
- Universal Message Translator
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- Universal Message Service (OMS)
- Universal Controller Command Line Interface (CLI)
- System Files
- Universal Translate Tables (UTT)
- Python Distribution for Universal Agent
    - Provided Modules

Universal Agent for Windows - File Inventory Lists

The Universal Agent 7.2.x for Windows System Installation package includes the files required for the following components / utilities:

- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Configuration Manager
- Universal Connector for PeopleSoft
- Universal Connector for SAP
- Universal Control Manager
- Universal Control Server
- Universal Copy
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Encrypt
- Universal Event Log Dump
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- Universal Message Service (OMS)
- Universal Controller Command Line Interface (CLI)
- Python Distribution for Universal Agent

Universal Broker and Universal Configuration Manager always are installed. Other components are installed as desired. If any of the components already are installed, Windows Installer will upgrade them to the latest version.

This page lists the files installed with each Universal Agent for Windows component. The file paths specified are relative to the root installation directory (for example, **C:\ProgramFiles\Universal**) that was specified during the installation, except **%ALLUSERSPROFILE%**, which resolves as documented.

Items shown with a path of System32 are installed in the Windows system folder. The actual name of this directory depends on the Windows version

- For all supported 64-bit Windows editions when the 64-bit Universal Agent package is installed, the path is **\Windows\System32**.

Universal Automation Center Agent

File	Description
UAGSrv\bin\uagsrv.exe	Universal Automation Center Agent Application program.
UAGSrv\bin\ops_copyfile.exe	Universal Automation Center Agent copy utility.
UAGSrv\bin\ops_scan.exe	Universal Automation Center Agent scan utility.
UAGSrv\bin\opsmerge.vbs	Visual Basic script used to migrate an existing 1.5, 1.6, or 1.7 Opswise Agent to UAG.
UAGSrv\bin\uftp.exe	Universal FTP Client program.
UAGSrv\samples	UAG sample directory.
UAGSrv\samples\OPSWISE-MIB.txt	MIB file.
UAGSrv\samples\parser.txt	Universal FTP Client file describing the format of file listings received from FTP servers, based on operating system and/or file system.
UBroker\tmpl\uagcmp	Template file for the Universal Automation Center Agent component definition.
UBroker\tmpl\uagcfg	Template file for the Universal Automation Center Agent configuration.
UBroker\tmpl\uagcmp	Template file for the Universal FTP Client component definition.
nls\uagmceng.umf	English message catalog.
Universal\cmdref	Universal Agent command Reference files directory
Universal\keystore	Broker Key Store.
%ALLUSERSPROFILE% \Universal\conf\uags.conf	Universal Automation Center Agent configuration file.
%ALLUSERSPROFILE% \Universal\comp\uag	Universal Automation Center Agent component definition file.
%ALLUSERSPROFILE% \Universal\conf\uftp.conf	Universal FTP Client configuration file.

Universal Broker

File	Description
UBroker\bin\ubroker.exe	Console application program.
UBroker\bin\ubrsvc.exe	Windows service program.
UBroker\bin\ubrdbrec.bat	Recovers Universal Agent database files.
UBroker\bin\xpsmsgnt.dll	Universal Broker file, used to interface with the Windows Application Event Log, containing IDs for the messages that Universal Broker may display.
UBroker\tmpl	XML template files used by I-Management Console for remotely configuring Universal Agent.
UBroker\tmpl\aclcfg	Universal Access Control List (UACL) configuration template.
UBroker\tmpl\ubrcfg	Universal Broker configuration template.
nls\ussmceng.umc	English message catalog.
%ALLUSERSPROFILE% \Universal\conf\uacl.conf	Universal Access Control List (ACL) configuration file **.%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\DocumentsandSettings\All Users**.
%ALLUSERSPROFILE% \Universal\conf\ubroker.conf	Universal Broker configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\DocumentsandSettings\AllUsers**.

Universal Command Manager

File	Description
System32\ucmd.exe	Application program.

UBroker\tmpl\ucmcfg	Universal Command Manager configuration template.
nls\ucmmceng.umc	English message catalog.
%ALLUSERSPROFILE% \Universal\conf\ucmd.conf	Universal Command Manager configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\DocumentsandSettings\AllUsers**.

Universal Command Server

File	Description
UBroker\tmpl\ucscfg	Universal Command Server configuration template.
UBroker\tmpl\ucmcmp	Universal Command Server component definition template.
UCmdSrv\bin\ucmsrv.exe	Application program.
System32\urc.exe	Universal Return Code utility.
nls\ucmmceng.umc	English message catalog.
%ALLUSERSPROFILE% \Universal\comp\ucmd	Universal Command Server component definition file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\DocumentsandSettings\AllUsers**.
%ALLUSERSPROFILE% \Universal\conf\ucmds.conf	Universal Command Server configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\DocumentsandSettings\AllUsers**.

Universal Connector for PeopleSoft

File	Description
nls\uppmceng.umc	English message catalog.
UBroker\tmpl\uppcfg	Universal Connector for PeopleSoft configuration template.
UPPS\bin\upps.exe	Application program.
%ALLUSERSPROFILE%\Universal\conf\upps.conf	Universal Connector for PeopleSoft configuration file.

Universal Connector for SAP

File	Description
UBroker\tmpl\uspcfg	Universal Connector for SAP configuration template.
USap\bin\usap.exe	Universal Connector application program.
USap\samp\sapnwrfc.ini	SAP NW (NetWeaver) RFC interface configuration file in the sample file directory.
UBroker\tmpl\uspcfg	Universal Connector remote configuration template.
nls\uspmceng.umc	Universal Connector English message catalog.
nls*.stt	Spoollist translation table files.
%ALLUSERSPROFILE%\Universal\conf\usap.conf	Universal Connector configuration file.

Universal Control Manager

File	Description
System32\uctl.exe	Application program.
UBroker\tmpl\uctcfg	Universal Control Manager configuration template.
nls\uctmceng.umc	English message catalog.

%ALLUSERSPROFILE%\Universal\conf\uctl.conf	Universal Control Manager configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Control Server

File	Description
UBroker\tmpl\utscfg	Universal Control Server configuration template.
UBroker\tmpl\uctcmp	Universal Control Server component definition template.
UCtlSrv\bin\uctsrv.exe	Application program.
nls\uctmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Universal\comp\uctl	Universal Control Server component definition file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.
%ALLUSERSPROFILE%\Universal\conf\uctls.conf	Universal Control Server configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Data Mover Manager

File	Description
System32\udm.exe	Application program.
UBroker\tmpl\udmcfg	Universal Data Mover Manager configuration template.
nls\udmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Universal\conf\udm.conf	Universal Data Mover Manager configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Data Mover Server

File	Description
UBroker\tmpl\udscfg	Universal Data Mover Server configuration template.
UBroker\tmpl\udmcmp	Universal Data Mover Server component definition template.
UdmSrv\bin\udmsrv.exe	Application program.
nls\udmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Universal\comp\udm	Universal Data Mover Server component definition file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.
%ALLUSERSPROFILE%\Universal\conf\udms.conf	Universal Data Mover Server configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Event Monitor Manager

File	Description
System32\uem.exe	Application program.
UBroker\tmpl\uemcfg	Universal Event Monitor Manager configuration template.
nls\uemmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Universal\conf\uem.conf	Universal Event Monitor Manager configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Event Monitor Server

File	Description
System32\uemload.exe	Event definition and event handler database load utility.
UBroker\tmpl\uedcmp	Universal Event Monitor Demand-Driven Server component definition template.
UBroker\tmpl\uemcmp	Universal Event Monitor Event-Driven Server component definition template.
UBroker\tmpl\uescfg	Universal Event Monitor Server configuration template.
UemSrv\bin\uemsrv.exe	Application program.
nls\uemmceng.umc	English message catalog.
%ALLUSERSPROFILE% \Universal\comp\uemd	Component definition file for the Universal Event Monitor Demand-Driven Server. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.
%ALLUSERSPROFILE% \Universal\comp\uems	Component definition file for the Universal Event Monitor Event-Driven Server. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.
%ALLUSERSPROFILE% \Universal\conf\uems. conf	Universal Event Monitor Server configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Configuration Manager

File	Description
UCfgMgr\bin\ucfgmgr.cpl	Universal Configuration Manager control panel application.
UCfgMgr\bin\UCfgMgrHelp.pdf	Universal Configuration Manager help file.

Universal Copy

File	Description
System32\ucopy.exe	Utility used for binary file copies. Similar to the UNIX cat command. Installed only if Universal Command Server is installed.

Universal Encrypt

File	Description
System32\uencrypt.exe	Application program. Installed only if Universal Command Manager is installed.

Universal Event Log Dump

File	Description
System32\ueld.exe	Application program.
UBroker\tmpl\uelcfg	Universal Event Log Dump configuration template.
nls\uelmceng.umc	English message catalog.
%ALLUSERSPROFILE% \Universal\conf\ueld.conf	Universal Event Log Dump configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Message Translator

File	Description
System32\umet.exe	Application program, always installed. Used to map application-specific error messages to error codes.

Universal Products Install Merge

File	Description
UPIMerge\bin\upimerge.exe	Application program, always installed. Provides command line access to the same functionality used by the Universal Agent installation to merge options from a new configuration file into an existing file.

Universal Query

File	Description
System32\uquery.exe	Application program.
UBroker\tmpl\uqrcfg	Universal Query configuration template.
nls\uqrmceng.umc	English message catalog.
%ALLUSERSPROFILE\Universal\conf\uquery.conf	Universal Query configuration file. **%ALLUSERSPROFILE%** is a system environment variable that expands to the All Users profile directory, typically **C:\Documents and Settings\All Users**.

Universal Spool Utilities

File	Description
USpool\bin\uslist.exe	Used to list the contents of Universal Spool files.
USpool\bin\uslrm.exe	Used to remove records from Universal Spool files.
USpool\bin\udb_archive.exe **USpool\bin\udb_dump.exe** **USpool\bin\udb_load.exe** **USpool\bin\udb_recover.exe** **USpool\bin\udb_stat.exe** **USpool\bin\udb_upgrade.exe** **USpool\bin\udb_verify.exe**	Miscellaneous spool file utilities. Should be used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support.

Universal Message Service (OMS)

File	Description
omssrv\bin\omssrv.exe	OMS Server.
omssrv\bin\omsadm.exe	OMS Administration utility.
nls\omsmceng.umc	OMS message catalog.
UBroker\tmpl\omscfg	OMS Server configuration template.
UBroker\tmpl\omscmp	OMS Server component definition template.
%ALLUSERSPROFILE%\Universal\conf\omss.conf	OMS configuration file.
%ALLUSERSPROFILE%\Universal\comp\oms	OMS component definition file.

Universal Controller Command Line Interface (CLI)

File	Description
OpsCli\bin\uagcmd	CLI executable command.
OpsCli\bin\ops-change-credentials-password.bat	Changes the runtime password for the specified Universal Controller credentials resource.
OpsCli\bin\ops-change-user-password.bat	Changes the password for the specified Universal Controller User account.
OpsCli\bin\ops-connector-status.bat	Lists the status of one or more Connectors.
OpsCli\bin\ops-export-bulk.bat	Performs a bulk export of all Controller database records.
OpsCli\bin\ops-export-trigger.bat	Performs an export of the specified triggers and any associated records.
OpsCli\bin\ops-import-bulk.bat	Imports Controller database records that were exported with ops-export-bulk.
OpsCli\bin\ops-import-trigger.bat	Imports triggers and associated records that were exported using ops-export-trigger.
OpsCli\bin\ops-manual-setcompleted.bat	Sets Manual task status to Success.
OpsCli\bin\ops-manual-setstarted.bat	Changes Manual task status from Action Required to Started.
OpsCli\bin\ops-resume-agent.bat	Allows a suspended Agent to submit tasks once again.
OpsCli\bin\ops-resume-agent-cluster.bat	Allows a suspended Agent Cluster to execute tasks once again.
OpsCli\bin\ops-resume-agent-cluster-membership.bat	Allows a specified Agent to rejoin specified Agent Cluster.
OpsCli\bin\ops-set-agent-cluster-task-execution-limit.bat	Sets the task execution limit for an Agent Cluster.
OpsCli\bin\ops-set-agent-task-execution-limit.bat	Sets the task execution limit for an Agent.
OpsCli\bin\ops-suspend-agent.bat	Temporarily prevents a specified Agent from submitting tasks.
OpsCli\bin\ops-suspend-agent-cluster.bat	Temporarily prevents the Agents in the specified Cluster from submitting tasks.
OpsCli\bin\ops-suspend-agent-cluster-membership.bat	Temporarily removes an Agent from cluster membership.
OpsCli\bin\ops-task-cancel.bat	Cancels a task.
OpsCli\bin\ops-task-forcefinish.bat	Force Finishes a task.
OpsCli\bin\ops-task-hold.bat	Places a task on hold.
OpsCli\bin\ops-task-launch.bat	Executes a task.
OpsCli\bin\ops-task-list.bat	Lists the specified tasks.
OpsCli\bin\ops-task-release.bat	Releases a held task.
OpsCli\bin\ops-task-rerun.bat	Re-executes the specified task.
OpsCli\bin\ops-task-setpriority.bat	Changes the execution priority of a Started task.
OpsCli\bin\ops-task-skip.bat	Skips the specified task instance.
OpsCli\bin\ops-task-status.bat	Displays the status of the task instance(s) associated with a task.
OpsCli\bin\ops-trigger-disable.bat	Disables the specified trigger(s).
OpsCli\bin\ops-trigger-enable.bat	Enables the specified trigger(s).
OpsCli\bin\ops-trigger-now.bat	Satisfies all conditions in the specified trigger and launches its associated tasks.
OpsCli\bin\ops-trigger-status.bat	Displays the status of the specified trigger(s).
OpsCli\bin\ops-update-resource-limit.bat	Sets the resource limit for a resource.
OpsCli\bin\ops-variable-list.bat	Displays the specified variable's value.
OpsCli\bin\ops-variable-set.bat	Sets the specified variable's value.
OpsCli\bin\uagcmd.exe	CLI Executable binary.

%ALLUSERSPROFILE%\Universal\conf\cmdtools.props	Optional CLI configuration file.

System Files

The following files will be installed only if they are newer than the existing file.

All files are installed to the Windows System32 folder, unless otherwise noted.

Description	File
Visual C++ 14 Runtimes	concrt140.dll msvcp140.dll msvcp140_1.dll msvcp140_2.dll ucrtbase.dll vccorlib140.dll vcruntime140_1.dll
Microsoft Foundation Class 14.0	mfc140.dll mfc140u.dll mfcm140.dll mfcm140u.dll mfc140chs.dll mfc140cht.dll mfc140deu.dll mfc140enu.dll mfc140esn.dll mfc140fra.dll mfc140ita.dll mfc140jpn.dll mfc140kor.dll mfc140rus.dll
Microsoft Universal C Runtime [1]	api_ms_win_core_console_l1_1_0.dll api_ms_win_core_datetime_l1_1_0.dll api_ms_win_core_debug_l1_1_0.dll api_ms_win_core_errorhandling_l1_1_0.dll api_ms_win_core_file_l1_1_0.dll api_ms_win_core_file_l1_2_0.dll api_ms_win_core_file_l2_1_0.dll api_ms_win_core_handle_l1_1_0.dll api_ms_win_core_heap_l1_1_0.dll api_ms_win_core_interlocked_l1_1_0.dll api_ms_win_core_libraryloader_l1_1_0.dll api_ms_win_core_localization_l1_2_0.dll api_ms_win_core_memory_l1_1_0.dll api_ms_win_core_namedpipe_l1_1_0.dll api_ms_win_core_processenvironment_l1_1_0.dll api_ms_win_core_processthreads_l1_1_0.dll api_ms_win_core_processthreads_l1_1_1.dll api_ms_win_core_profile_l1_1_0.dll api_ms_win_core_rtlsupport_l1_1_0.dll api_ms_win_core_string_l1_1_0.dll api_ms_win_core_synch_l1_1_0.dll api_ms_win_core_synch_l1_2_0.dll api_ms_win_core_sysinfo_l1_1_0.dll api_ms_win_core_timezone_l1_1_0.dll api_ms_win_core_util_l1_1_0.dll api_ms_win_crt_conio_l1_1_0.dll api_ms_win_crt_convert_l1_1_0.dll api_ms_win_crt_environment_l1_1_0.dll api_ms_win_crt_filesystem_l1_1_0.dll api_ms_win_crt_heap_l1_1_0.dll api_ms_win_crt_locale_l1_1_0.dll api_ms_win_crt_math_l1_1_0.dll api_ms_win_crt_multibyte_l1_1_0.dll api_ms_win_crt_private_l1_1_0.dll api_ms_win_crt_process_l1_1_0.dll api_ms_win_crt_runtime_l1_1_0.dll api_ms_win_crt_stdio_l1_1_0.dll api_ms_win_crt_string_l1_1_0.dll api_ms_win_crt_time_l1_1_0.dll api_ms_win_crt_utility_l1_1_0.dll ucrtbase.dll

Microsoft Windows Installer 3.1.4000.1823 [2]	msiexec.exe

[1] The Microsoft Universal C Runtime is built into Windows 10, Server 2016, and Server 2019. It is available as a Microsoft update on Windows 7, Server 2008 R2, Server 2012, and Server 2012 R2 systems. Although the Universal Agent install provides these .dlls on older system, the install will issue an error and stop without installing any files if they do not already exist. The install executes functions that rely on the MS UCRT.

[2] See the documentation regarding the Microsoft Windows Installer.

Universal Translate Tables (UTT)

File
C:\Program Files\Universal\nls\default.stt
C:\Program Files\Universal\nls\raw.stt
C:\Program Files\Universal\nls\iso8859-10.utt
C:\Program Files\Universal\nls\iso8859-13.utt
C:\Program Files\Universal\nls\iso8859-14.utt
C:\Program Files\Universal\nls\iso8859-15.utt
C:\Program Files\Universal\nls\iso8859-1.utt
C:\Program Files\Universal\nls\iso8859-2.utt
C:\Program Files\Universal\nls\iso8859-3.utt
C:\Program Files\Universal\nls\iso8859-4.utt
C:\Program Files\Universal\nls\iso8859-5.utt
C:\Program Files\Universal\nls\iso8859-6.utt
C:\Program Files\Universal\nls\iso8859-7.utt
C:\Program Files\Universal\nls\iso8859-8.utt
C:\Program Files\Universal\nls\iso8859-9.utt
C:\Program Files\Universal\nls\pc437.utt
C:\Program Files\Universal\nls\pc737.utt
C:\Program Files\Universal\nls\pc775.utt
C:\Program Files\Universal\nls\pc850.utt
C:\Program Files\Universal\nls\pc852.utt
C:\Program Files\Universal\nls\pc855.utt
C:\Program Files\Universal\nls\pc857.utt
C:\Program Files\Universal\nls\pc860.utt
C:\Program Files\Universal\nls\pc861.utt
C:\Program Files\Universal\nls\pc862.utt
C:\Program Files\Universal\nls\pc863.utt
C:\Program Files\Universal\nls\pc864.utt
C:\Program Files\Universal\nls\pc865.utt
C:\Program Files\Universal\nls\pc866.utt
C:\Program Files\Universal\nls\pc869.utt
C:\Program Files\Universal\nls\pc874.utt
C:\Program Files\Universal\nls\win1250.utt

C:\Program Files\Universal\nls\win1251.utt
C:\Program Files\Universal\nls\win1252.utt
C:\Program Files\Universal\nls\win1253.utt
C:\Program Files\Universal\nls\win1254.utt
C:\Program Files\Universal\nls\win1255.utt
C:\Program Files\Universal\nls\win1256.utt
C:\Program Files\Universal\nls\win1257.utt
C:\Program Files\Universal\nls\win1258.utt

Python Distribution for Universal Agent

Installed under **.\Universal\Python**.

The Python distribution provided by the Universal Agent install is a fully-functional Python distribution that has complete support for Python script execution, virtual environment creation, and/or add-on module installations via pip.

When installing new Python modules to this distribution's install directory, remember to use the provided pip binary to ensure that the module is installed to the correct path.

Provided Modules

The modules provided by the Universal Agent's Python distribution are listed below. Where possible, the dependencies for each of these modules are also provided.

Stonebranch does not own, support, or maintain any of these modules. The link for each module will direct you to that project's homepage.

Module	Version	Notes
azure-storage-blob	12.7.1	
botocore	1.14.8	
boto3	1.11.8	
docker	4.1.0	
google-cloud-storage	1.25.0	
paramiko	2.7.1	
requests	2.22.0	
hdbcli	2.7.23	
pyodbc	4.0.30	
cx-Oracle	8.0.1	
hdfs	2.5.8	
kubernetes	12.0.1	
wheel		No version requirement enforced at build time

Universal Agent for Windows User Mode Installation

Introduction

The following information is provided for the user mode installation of Universal Agent for Windows:

- Installation Package
- Installation Requirements
- Installation Procedures
- File Inventory Lists

(For licensing information, see Windows Installation - Licensing.)

Universal Agent for Windows - User Mode Installation Package

- Components
- Component Compatibility
- Broker Service Management Commands
  - ubrsvc.bat
- Executing Manager Components and Utilities

Components

The Universal Agent for Windows package includes the following product components:

- Universal Broker 7.2.x
- Universal Automation Center Agent 7.2.x
- Universal Certificate 7.2.x
- Universal Command Manager and Server 7.2.x
- Universal Connector for SAP 7.2.x
- Universal Connector for PeopleSoft 7.2.x
- Universal Control Manager and Server 7.2.x
- Universal Controller Command Line Interface (CLI) 7.2.x
- Universal Data Mover Manager and Server 7.2.x
- Universal Encrypt 7.2.x
- Universal Event Log Dump 7.2.x
- Universal Event Monitor Manager and Server 7.2.x
- Universal Message Service (OMS)7.2.x
- Universal Message to Exit Code Translator 7.2.x
- Universal Query 7.2.x

Component Compatibility

The following table identifies the compatibility of Universal Agent for Windows components with previous component / product versions.

Component	Compatibility
Universal Broker 7.2.x	Universal Agent / Workload Automation / Stonebranch Solutions / Universal Products releases 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Command 7.2.x	Universal Command 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Control 7.2.x	Universal Control 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Data Mover 7.2.x	Universal Data Mover 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1 and 3.1.0.
Universal Query 7.2.x	Universal Broker 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Event Monitor 7.2.x	Universal Event Monitor 7.1.0, 7.0.0, 6.9.0, 6.8.0, 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

Broker Service Management Commands

`ubrsvc.bat` wraps several commands to control the Universal Broker service (that is, -start, -stop, -restart, -status, -remove). The `ubrsvc.bat` script is generated by the user mode installation and contains commands that target the Universal Broker service created and registered by that installation.

ubrsvc.bat

`ubrsvc.bat` supports the following command line syntax:

```
ubrsvc { -start | -stop | -restart | -status | -remove } [-[h]elp]
```

`-start`	Starts the Broker service.
`-stop`	Stops the Broker service.
`-restart`	Stops and starts the Broker service.
`-status`	Reports the current status of the Broker.
`-remove`	Stops the Broker service, removes it from the Service Control Manager (SCM), and unregisters its message source. Important Be sure to run the provided **`SB_ConfigSetup.bat`** script before removing the Universal Broker service definition. Doing so will ensure that the correct Broker is removed when multiple user mode installs exist.
`-h`	Displays this message.

Note

The `-remove` option is a critical first step when removing a user mode installation.

Executing Manager Components and Utilities

When executing a Universal Agent Manager component (for example, Universal Command Manager or Universal Data Mover Manager), from a user mode installation, execute `SB_ConfigSetup.bat` to configure the environment for proper local Universal Broker registration.

Failure to do this may cause a component of one version to register with the system-installed Universal Broker, which could be a different version.

This script should also be run prior to using the **`-remove`** option to ubrsvc.bat. See note above.

Universal Agent for Windows - User Mode Installation Requirements

- Windows Versions
- Additional Requirements
- Platform Requirements
- Third-Party Requirements
- Installation Account
- Universal Broker Service
  - Selecting a Universal Broker Service Account
- Spool Directory
  - Universal Command Server
  - UAG Cache

Windows Versions

To install Universal Agent for Windows, you must have one of the following versions of Windows:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

Note

The Universal Agent for Windows User Mode installation package will execute only on a 64-bit version of Windows.

Additional Requirements

- Administrative authority to execute the installation.
- Network connectivity via TCP/IP.
- About 120 MB available disk space.
- Interactive access to the Windows system (command line support will be provided in an upcoming release).
- Microsoft Visual C++ 14 Runtime
- Microsoft Universal C Runtime

Note

The Universal Agent User Mode install does not provide the required Microsoft runtimes. If either of these do not exist, the install will issue an error and stop without installing any files. Both runtimes are available as redistributable packages from Microsoft. The Universal C Runtime is built into Windows 10, Server 2016, and Server 2019. It is available via a Microsoft update for Windows Server 2012 and Windows Server 2012 R2.

Platform Requirements

Since platform requirements may change with new releases of Universal Agent, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Third-Party Requirements

Some Universal Agent components may rely on vendor-supplied resources whose runtime dependencies are not known to the Agent before execution. One such component is the Universal Connector for SAP (USAP), which relies on libraries provided by SAP. If the runtime dependencies for those libraries are not available on your Agent system, attempts to run USAP will fail. Where such dependencies exist, use information provided by the vendor to determine runtime requirements. For example, to review runtime requirements for the NetWeaver RFC SDK version 750, refer to the information on this page.

Installation Account

The account used to execute a Universal Agent for Windows user mode installation must have administrative privileges.

Universal Broker Service

The Universal Broker service runs either with an Administrative account or with the Local System account.

If you are using an Administrative account, the account must have the following privileges on the system upon which the Universal Agent resides (optional, but recommended privileges are noted):

- Act as part of the operating system
- Adjust memory quotas for a process
- Bypass traverse checking
- Debug programs
- Deny log on as batch job (optional, recommended)
- Deny log on locally (optional, recommended)
- Deny log on through Terminal Services (optional, recommended)
- Log on as a service
- Impersonate a client after authentication
- Increase scheduling priority
- Replace a process level token
- Take ownership of files and other objects

Selecting a Universal Broker Service Account

The Universal Agent for Windows user mode install will supply default credentials for an account named UBrokerService. This is the same default account that the Universal Agent for Windows system installation creates (if it does not already exist).

If the Universal Broker service will execute with a different account, you can provide that account's user ID and password during the install or specify them post-install in the Universal Broker service's configured settings within the Windows Services application.

If the Universal Broker service will execute with a domain account, that account must have the privileges listed above on the local Universal Agent system. The privileges can be granted locally or inherited via Active Directory policy, but they must be granted on the local system.

In all cases, the account chosen to run the Universal Broker service must be an existing Administrative account. The Universal Agent for Windows user mode install will not create it.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Agent components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with Manager Fault Tolerance enabled.
- Configuration information for Universal Agent components, when a local Universal Broker is operating in managed mode.

For user mode installations, the spool directory resides just under the main install directory.

- Most database files reside in `.\spool\ubroker`.
- Spooled standard I/O files reside in `.\spool\ucmdsrv`.

Universal Command Server

Location

The spool directory must reside on a local device. It cannot reside on any network device, including network drives that may be mapped to a local drive. By default, the spool files are located in directory `.\spool\ucmdsrv` (relative to the main installation directory).

Space

You must have approximately 125 MB available disk space for each instance of a Universal Agent for Windows user mode installation.

The amount of disk space required for the spool directory depend on the following factors:

- Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
- When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
- Average size of the user processes standard input, output, and error files. Keep in mind that spooling is not a feature for file transfer purposes. File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

```
MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = required disk space.
```

For example, if the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. This feature can be enabled by updating the Universal Command Server configuration file, `.\conf\ucmds.conf`.

For more information on the Manager Fault Tolerant feature and spooling of redirected standard I/O files, see the Universal Agent 7.2.x User Guide.

Security

Universal Broker and Universal Command Server require read/write access to the spool directory. No other Universal Agent components access the spool directory directly.

UAG Cache

The Universal Automation Center Agent (UAGSRV) stores standard I/O redirected from processes it executes to files in the UAG cache locations below.

Space

Cache files are located, by default, in directory `.\UagSrv\cache`, relative to the main Universal Agent for Windows user mode installation directory.

Cache files are created for each job that is run by Universal Agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by Universal Controller. You can configure the number of days that files remain in the cache using the Agent Cache Retention Period in Days Universal Controller system property.

For each instance of a Universal Agent user mode installation, the amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Controller daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.`

For example:

If the files are purged every 7 days, and you run 1200 jobs daily on a particular user mode Universal Agent and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

UAG automatically redirects the standard output and standard error files to the cache directory with no required user input.

Security

Universal Automation Center Agent (UAG) requires read/write access to the UAG cache directory. No other Universal Agent components access the cache directory. No general user access is required.

Universal Agent for Windows - User Mode Installation Procedures

Installation Procedures

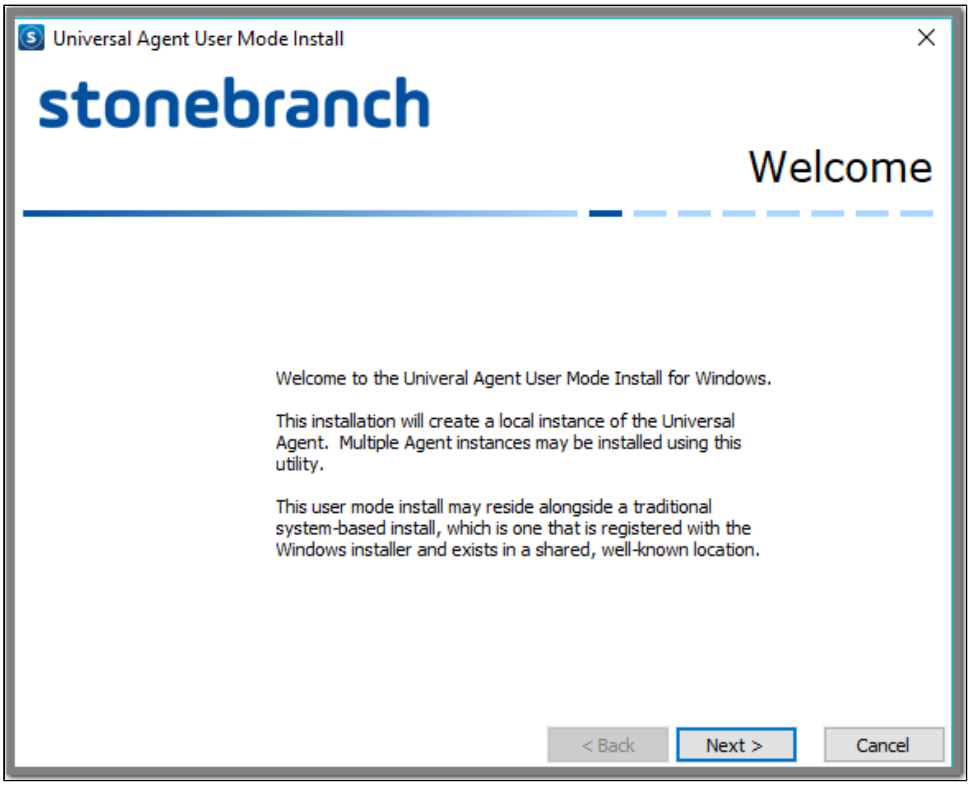
The following procedures are provided for the a Universal Agent for Windows user mode installation and removal:

- Installing a Universal Agent User Mode Install via the Graphical Interface
- Installing a Universal Agent User Mode Install via the Command Line Interface
- Removing a Universal Agent User Mode Install

Installing a Universal Agent User Mode Install via the Graphical Interface

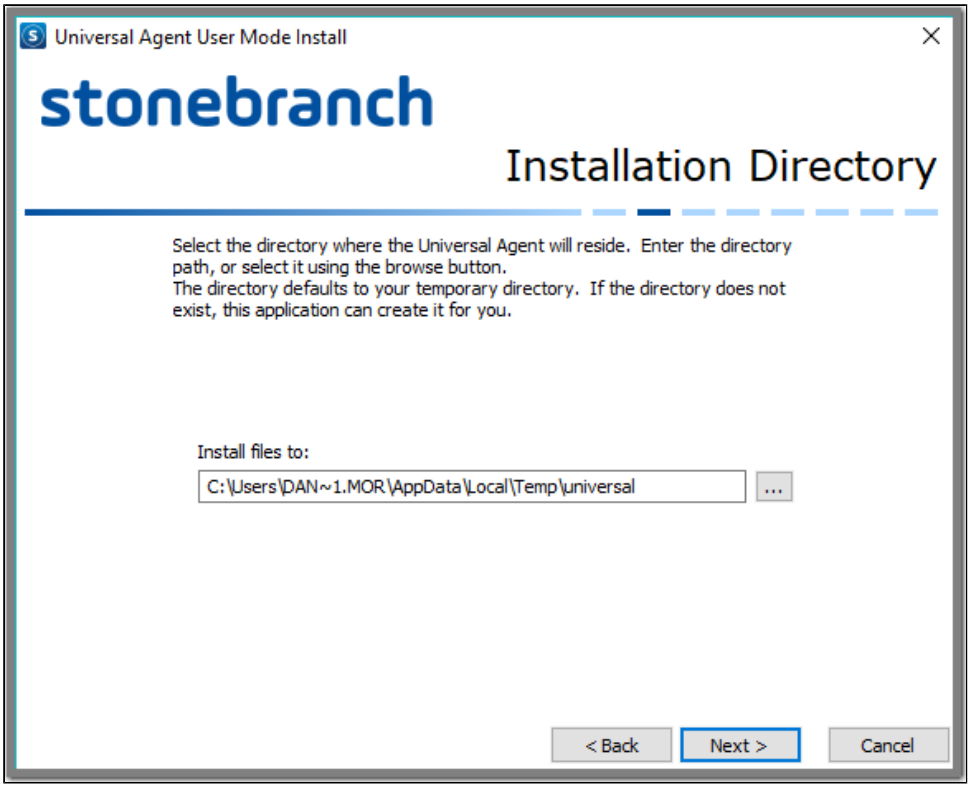
Installing a Universal Agent for Windows User Mode install via the Windows Installer Graphical Interface

To install a Universal Agent for Windows User Mode Install via the Windows Installer graphical interface, perform the following steps.

Step 1	Download the desired Universal Agent for Windows user mode installation file to your work station: • `sb-7.2.<mod>.<level>-windows-x64-um.exe`, the 64-bit Universal Agent for Windows user mode distribution file.
Step 2	Execute the distribution file. You can launch it from an Administrative command prompt or by double-clicking the file in Windows Explorer. When launched from Windows Explorer, you may be prompted for Administrative credentials so that the installation can execute with elevated privileges.
Step 3	When the installation starts, it will verify your Windows version to ensure it is a supported version. It also will look for the Microsoft VC++ 2010 runtime libraries, which are required to run the Universal Agent. If either of these checks fail, the installation will finish. If all installation pre-requisites are satisfied, the Welcome dialog displays. Click Next to proceed to the next dialog in the installation sequence. Select Back to return to any previous dialog. Select Cancel to terminate the installation (no system changes are made until the Progress dialog displays). 

Step 4

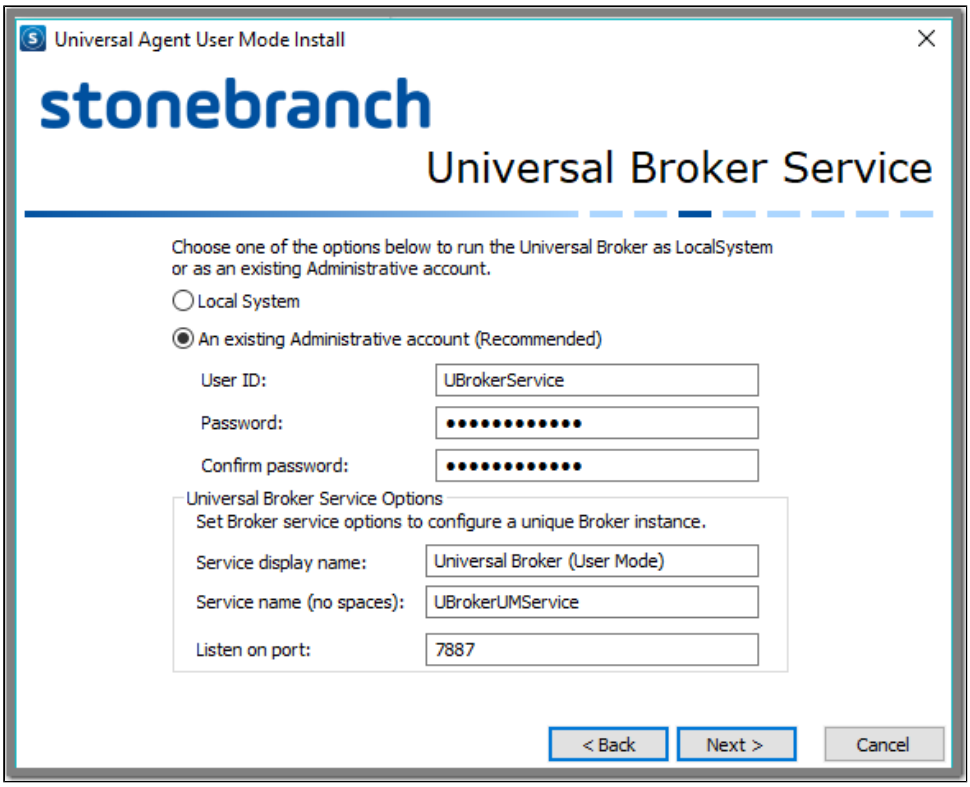
After the Welcome dialog, the Installation Directory dialog displays. Enter the desired installation location or select it using the **...** button. If the specified installation directory does not exist, the installation will create it.



In the example above, the user mode installation will reside under the `C:\Users\DAN~1.MOR\AppData\Local\Temp\universal` folder.

Step 5

Click the Next button to display the Universal Broker Service configuration dialog.



From this dialog, you can choose to run the Universal Broker service as LocalSystem or an existing Administrative account. If the account does not exist, the installation will not issue an error here. Instead, the installation will fail when it attempts to create and configure the Broker service.

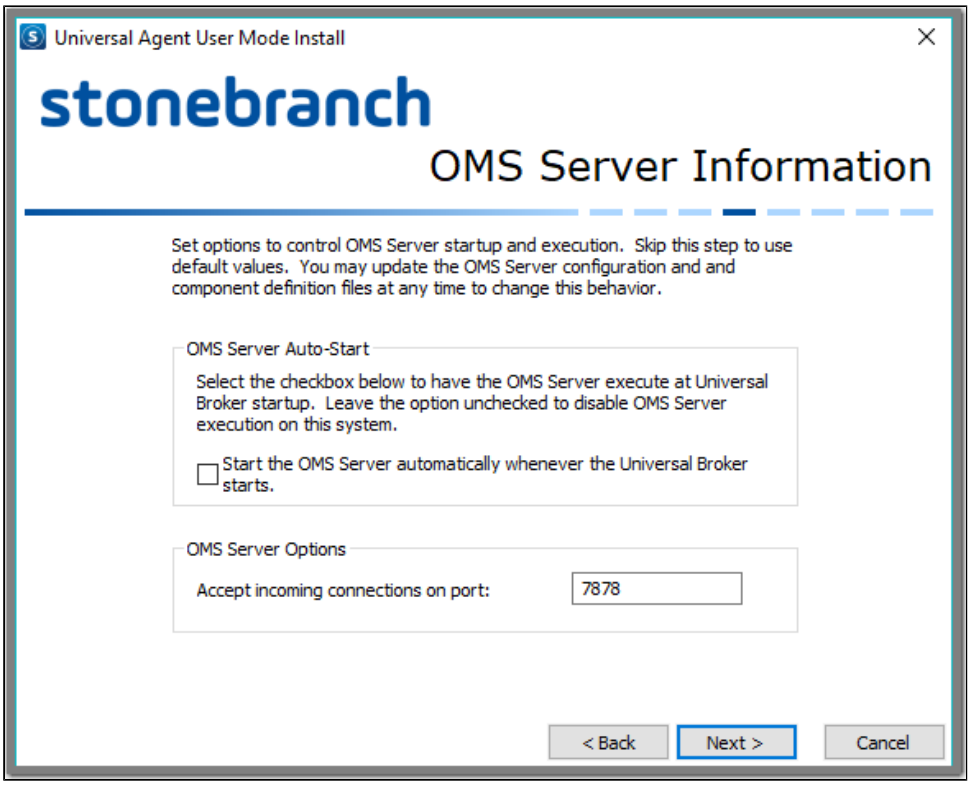
The Universal Broker service's display name and service name also can be specified from this dialog. The installation will ensure that the service name is unique.

The port upon which the Universal Broker service accepts incoming connections is specified in the **Listen on port** field. Note that the installation will not check whether another process is already accepting incoming connections over the specified port. It is your responsibility to ensure that the port is available.

The value specified here is used to set the **service_port** option in the Universal Broker configuration file, `.\conf\ubroker.conf`. 100 is added to this value to set the **component_port** option in `ubroker.conf`. The Broker accepts external connections over the **service_port** and uses the **component_port** to accept control sessions from local Universal Server components. Either of these values can be changed after the installation.

Step 6

Click the Next button to display the OMS Server Information dialog.

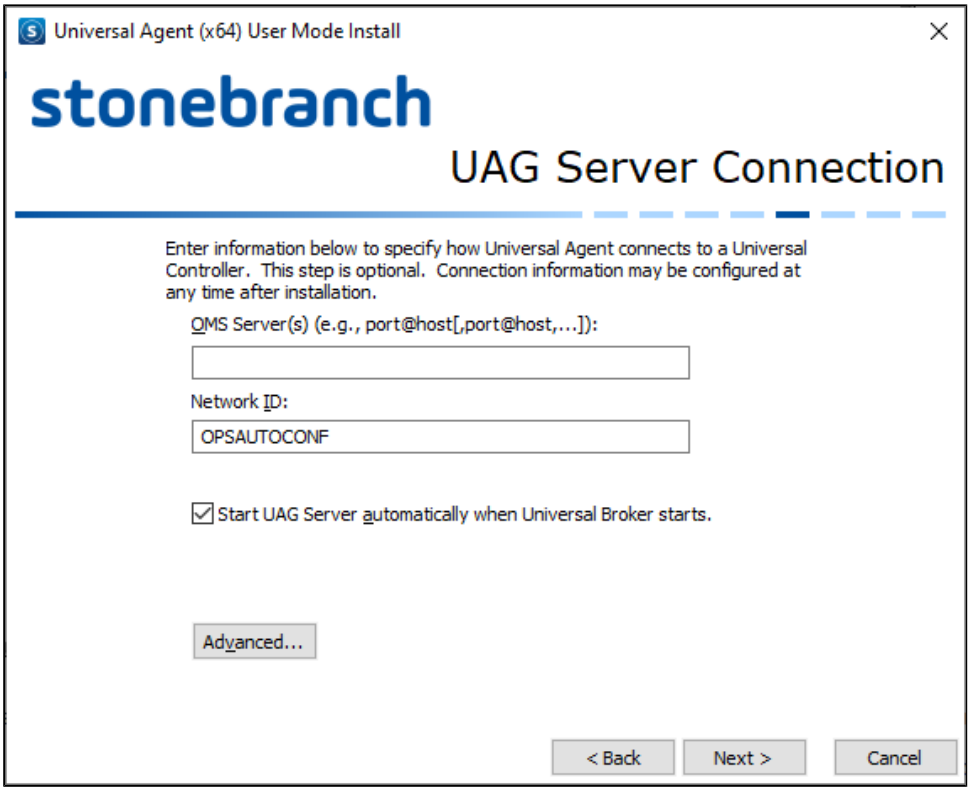


The option to automatically start the OMS Server is necessary only if local or remote Universal Agents will use this OMS Server to communicate with a Universal Controller. If the Universal Agent will connect using an existing OMS Server, the option can be left unchecked.

If the OMS Server will run from this installation, enter a port number upon which it will accept connections from local or remote Universal Agents. This port number must be unique, and like the Universal Broker port, the user mode installation makes no attempt to confirm its availability.

Step 7

Click the Next button to display the UAG Server Connection dialog.

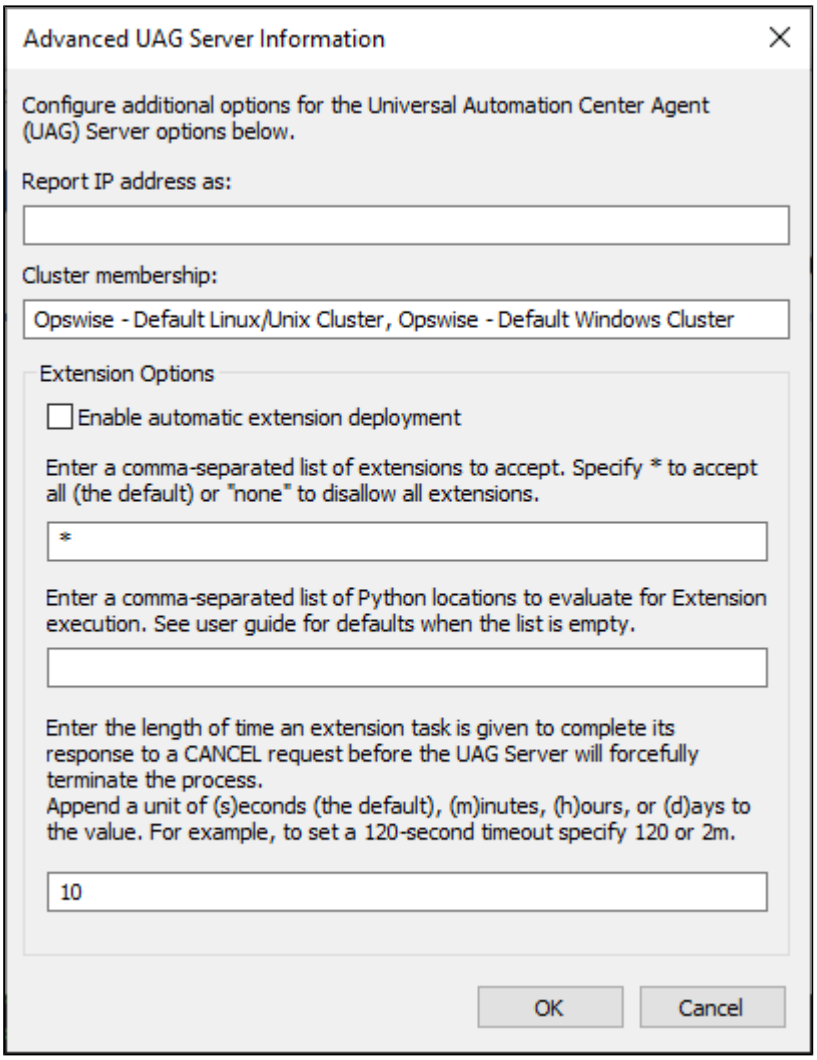


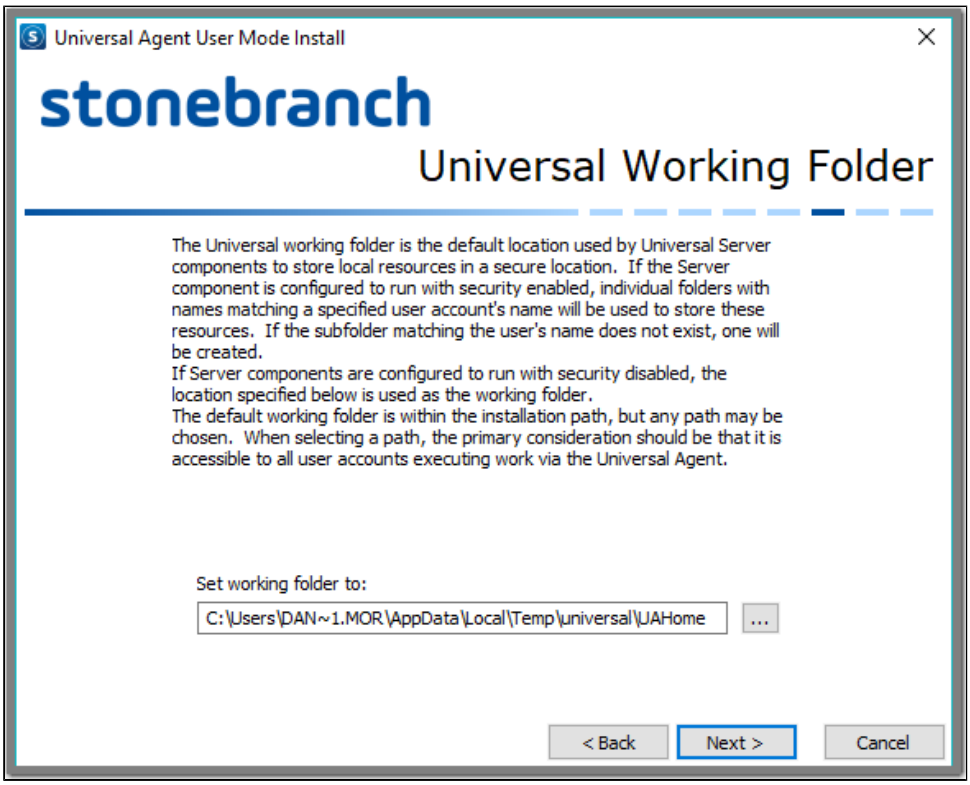
Use this dialog to configure how the UAG Server connects to and identifies itself to a Universal Controller.

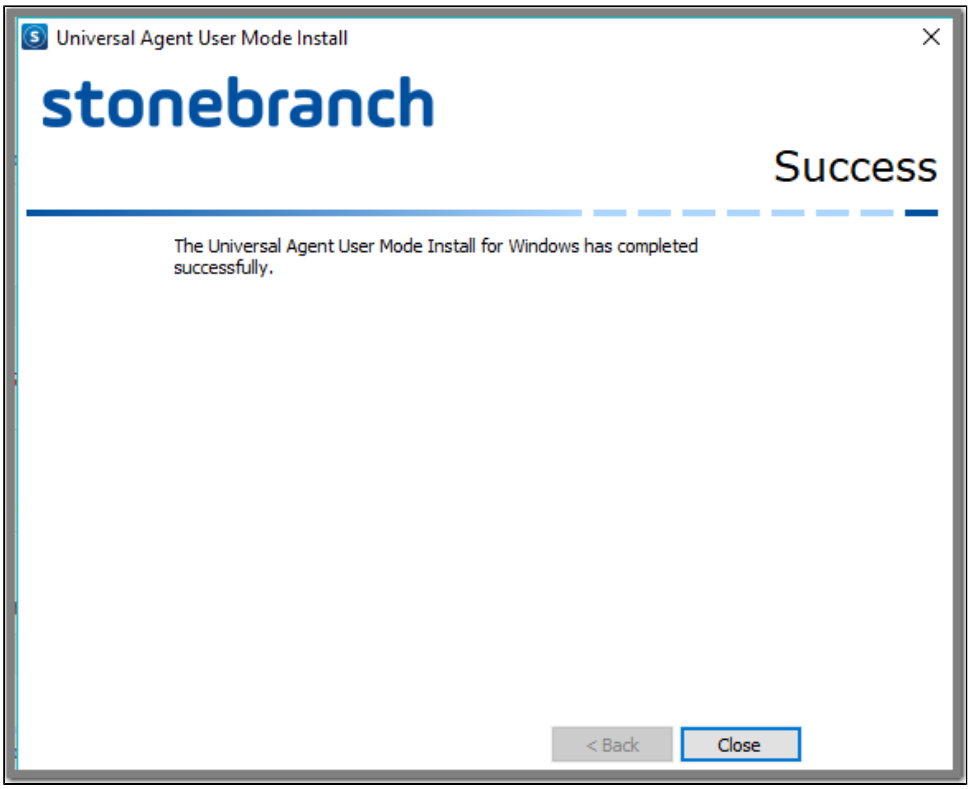
One or more local or remote OMS Servers can be specified. The OMS Server is the component responsible for exchanging messages between the Universal Controller and Universal Agent.

The Network ID controls how the Universal Agent is displayed in the Universal Controller. The default setting of **OPSAUTOCONF** instructs the Controller to assign a unique ID to the Agent. A specific value can be provided here to override this default and customize names for Universal Agents that the Universal Controller displays.

Click the **Advanced...** button to set additional UAG Server options.


Advanced UAG Server Information
Configure additional options for the Universal Automation Center Agent (UAG) Server options below.
Report IP address as:
Cluster membership:
Opswise - Default Linux/Unix Cluster, Opswise - Default Windows Cluster
Extension Options
Enable automatic extension deployment
Enter a comma-separated list of extensions to accept. Specify * to accept all (the default) or "none" to disallow all extensions.
*
Enter a comma-separated list of Python locations to evaluate for Extension execution. See user guide for defaults when the list is empty.
Enter the length of time an extension task is given to complete its response to a CANCEL request before the UAG Server will forcefully terminate the process.
Append a unit of (s)econds (the default), (m)inutes, (h)ours, or (d)ays to the value. For example, to set a 120-second timeout specify 120 or 2m.
10
OK
Cancel

Step 8	Click the Next button to display the Universal Working Folder dialog. This working folder is shared between the Universal Command Server, Universal Data Mover Server, and Universal Event Monitor Server components. If the directory does not exist, the installation will create it. 
Step 9	Click the Next button to begin the installation. The installation first unloads an embedded `.cab` file that contains the Universal Agent application files and resources. The installation extracts the contents of this `.cab` file to create the Universal Agent's application directory structure. After the files are placed on the file system, the installation uses values provided in the previous dialogs to configure the Universal Agent. The installation updates configuration and component definition files with these values. Next, the installation creates the Universal Broker service and configures it to run via the Windows Service Control Manager (SCM). The Broker's message source also is registered with the Windows Application event log. Finally, the installation creates two helper scripts: `SB_ConfigSetup.bat` and `ubrsvc.bat`. • `SB_ConfigSetup.bat` configures an environment to simplify execution of Universal Manager (for example: UCMD, UDM) and Universal Utility (for example: UENCRYPT, UFTP) components. Run this script to ensure that those components register with (and receive the correct configuration from) the local Universal Broker installed with the user mode installation. If this script is not run and a system-installed Universal Broker is running, the user mode-installed components may unintentionally register with it or fail local Broker registration altogether. • `ubrsvc.bat` wraps several commands to control the Universal Broker service (that is, -start, -stop, -restart, -status, -remove). The Broker service also can be controlled via the Windows SCM, but familiarity with the script is important, and useful particularly when unin stalling a user mode installation.

Step 10	When the installation finishes, the Complete dialog displays.  If any errors were encountered during the installation, this dialog is replaced by a Failed dialog.
Step 11	Select the Close button to exit the installation.
Step 12	Verify the results of the installation and start the Universal Broker service. The installation configures it to start automatically, but the service waits to be started for its initial run.

Installing a Universal Agent User Mode Install via the Command Line Interface

- Installing a Universal Agent for Windows User Mode Install via the Command Line Interface
- Installing the Agent
  - Command Line Syntax
  - Command Line Parameters
  - Command Line Installation Examples

Installing a Universal Agent for Windows User Mode Install via the Command Line Interface

This page describes how to install Universal Agent using the user mode install command line interface.

A command line installation is useful in situations where:

- Automation is used to deploy Universal Agent installations across many different systems.
- It is not practical or convenient to perform the graphical interface installation

Installing the Agent

To install Universal Agent for Windows using the user mode command line interface, perform the following steps:

Step 1	Download the Universal Agent for Windows User Mode Install product distribution file, **`sb-7.2.0.<level>-windows-x64-um.exe`**, to your work station, where **`<level>`** is the numeric package level.
Step 2	Execute the distribution file from the command line, and include all appropriate command line parameters.

Command Line Syntax

The following illustrates the command line syntax used to install the Agent:

`sb-7.2.0.<level>-windows-x64-um.exe`command line parameters

In this syntax:

- `<level>` is the numeric package level.

The command line parameters control the behavior of the install process. Where noted, some command line parameters have environment variable and configuration file equivalents that are available as alternate sources of user mode install input. The user mode install evaluates each source and combines the values to form a single set of input data. Environment variables override configuration file values, and command line parameter values override environment variables.

Command Line Parameters

The following table describes the parameters that are available for a command line installation.

The parameters can be specified in any order. The order of precedence for the different sources are:

1. Command line parameters (overrides all)
2. Command file parameters
3. Environment variables
4. Configuration file values

Parameter	Description	Syntax			Default
		Command Line	Environment Variable	Configuration File	

AC_AGENT_CLUSTERS=*clusters*	Sets the AGENT_CLUSTERS UAG configuration option to specify the Universal Controller-defined agent clusters to which this agent will belong.	-ac_agent_clusters *val*	UMIAGENTCLUSTERS=*val*	ac_agent_clusters *val*	**Opswise - Default Linux /Unix Cluster, Opswise - Default Windows Cluster**
AC_AGENT_IP=*IP address*	Sets the AGENT_IP UAG configuration option to have UAG Server register with Universal Controller using a specific IP address.	-ac_agent_ip *ipaddr*	UMIAGENTIP=*ipaddr*	ac_agent_ip *ipaddr*	(none)
AC_ENABLE_SSL	*This option is deprecated starting with Universal Agent 7.1.0.0. UAG Server will always attempt to use SSL/TLS for OMS connections.*	n/a	n/a	n/a	n/a
AC_EXTENSION_ACCEPT_LIST	Sets the EXTENSION_ACCEPT_LIST UAG configuration option, which is a comma-separated list of Universal Extensions the Agent will accept. The default value for this option is "*", which will cause the Agent to accept all extensions. Specify a value of "none" to cause the Agent to disallow execution of all Extensions.	-ac_extension_accept_list *list*	UMIEXTACCEPTLIST=*list*	ac_extension_accept_list	*****
AC_EXTENSION_CANCEL_TIMEOUT=*value***[s|m|h|d]**	Sets the EXTENSION_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time that a Universal Extension task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (s)econds, (m)inutes, (h)ours, or (d)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	-ac_extension_cancel_timeout *val*[**s|m|h|d**]	UMIEXTCANCELTIMEOUT=*val*[**s|m|h|d**]	ac_extension_cancel_timeout *val*[**s|m|h|d**]	10
AC_EXTENSION_DEPLOY_ON_REGISTRATION	Sets the EXTENSION_DEPLOY_ON_REGISTRATION UAG configuration option that controls whether UAG Server will automatically receive Universal Extensions from Universal Controller at Agent startup.	-ac_extension_deploy_on_registration *opt*	UMIEXTDEPLOYONREG=opt	ac_extension_deploy_on_registration *opt*	**NO**
AC_EXTENSION_PYTHON_LIST	Sets the EXTENSION_PYTHON_LIST UAG configuration option, which is a comma-separated list of all Python interpreters that should be considered for Universal Extension execution. There are some default locations the Agent will check even if this option is not specified. See the EXTENSION_PYTHON_LIST option's description for more information.	-ac_exension_python_list *list*	UMIEXTPYTHONLIST=*list*	ac_extension_python_list *list*	(none)
AC_NETNAME=*value*	Specifies a value for a network ID that uniquely identifies the Agent on this system to the Universal Controller. The install uses this value to set the UAG NETNAME configuration option.	-ac_netname *name*	UMIUAGSRVNETNAME *name*	ac_netname *name*	**OPSAUTOCONF**
BROKERDISPNAME	Specifies the name with which the Universal Broker for this user mode install is shown in the Windows Services Manager. This value is also used to register the message source for messages the Universal Broker sends to the Windows Application event log. Note that this is only the "friendly" name of the service, intended to uniquely identify the Broker service. The value specified for the BROKERSERVICENAME is the internal name the Windows Service Control Manager uses to control the service.	-broker_dispname *name*	UMIBROKERDISPNAME=*name*	broker_dispname *name*	**Universal Broker (User Mode)**
BROKERPORT	Specifies the port number or service name upon which the Universal Broker accepts connections from Universal Agent components. The value specified for this parameter must be unique for all services on the system, including other Universal Brokers.	-broker_port *port*	UMIBROKERPORT=*port*	broker_port *port*	**7887**

BROKERPWD=*BrokerAccountPassword*	Used by the install when RUNBROKERASUSER is **yes** to specify the password for the account used to execute the Universal Broker service. Valid values are any 256 characters or less. **BROKERPWD** is ignored if RUNBROKERASUSER is **no**.	-w *password* -broker_pwd *password*	UMIBROKERPWD=*password*	broker_password *password*	**BrSvcAcct001**
BROKERSERVICENAME	Specifies the name with which the Universal Broker is assigned and recognized by the Windows Services Control Manager. This is the "true" name of the service and is the value used for service control via the command line or Windows APIs. The value specified for this parameter must be unique among all the services – including other Universal Brokers – installed on the system.	-broker_svcname *name*	UMIBROKERSVCNAME=*name*	broker_svcname *name*	**UBrokerUMService**
BROKERUSERID	Used by the install when RUNBROKERASUSER is **yes** to set the user account used to execute the Universal Broker service. Valid values are any 20 characters or less. For domain accounts, a domain name of up to 256 characters is accepted, but the user ID is still limited to 20 characters. **BROKERUID** is ignored if RUNBROKERASUSER is **no**. Important The user account specified by this parameter MUST exist before the Broker can execute. The User Mode install will not create the account. Refer to the Universal Agent for Windows - User Mode Installation Requirements section for more information.	-u *userid* -broker_userid *userid*	UMIBROKERUID=*userid*	ubroker_userid *userid*	**UBrokerService**
COMMAND_FILE	Specifies a text file that contains command line parameters. This file may be used to simplify automated deployments and/or to shorten the length of the command line used to do the user mode installation. Note that this is not the same as a user mode installation CONFIGURATION_FILE. The format of each option specified within this file must follow *command line syntax*, and each option must reside on a separate line within the file. The path specified for the file may be an absolute path or one relative to the current working directory.	-f *path* -file *path* -	N/A	N/A	(none)
CONFIGURATION_FILE	Specifies a file in standard Universal Agent configuration file format that contains keywords and values that may be used to set installation parameters. Note that this file is not the same as a user mode installation COMMAND_FILE. The format of each option specified within this file must follow *configuration file syntax*, which each option place on a separate line within the file. The path specified for the file may be an absolute path or one relative to the current working directory.	-conf_file *path*	UMICONFIGFILE=*path*	N/A	(none)
HELP	Displays command line help and exits without installing. This parameter has no value associated with it.	-? -h -help	N/A	N/A	
INSTALLDIR	Sets the root user mode installation directory to *dir*. Each component will be installed under this directory. If the directory contains spaces, you must use double (**"**) quotation marks around the path name.	-installdir *dir*	UMIINSTALLDIR=*dir*	installation_directory *dir*	**%TEMP%\universal**
OMS_AUTOSTART	Specifies the behavior of the Universal Message Service (OMS) Server component at Universal Broker startup. • If **yes** is specified, the AUTOMATICALLY_START OMS component definition option is set to **yes**, which instructs Universal Broker to start the OMS Server automatically when it is started. • If **no** (the default) is specified, or if this option is not included on the command line, the AUTOMATICALLY_START and RESTART OMS component definition options are set to **no**, and Universal Broker will not start the OMS Server automatically when it is started. However, the OMS Server may still be started manually any time after the Broker is started via the Universal Agent's Universal Control utility.	-oms_autostart *opt*	UMIOMSAUTOSTART=*opt*	oms_autostart *opt*	**no**

OMS_PORT	Specifies the port number or service name upon which the Universal Message Service (OMS) Server accepts connections from the Universal Automation Center (UAG) Server component and the optional Universal Automation Center command line interface (UAGCMD). The value specified for this parameter must be unique for all services on the system, including other OMS Servers.	-oms_port *port*	UMIOMSPORT=*port*	oms_port *port*	**7878**
OMS_SERVERS	A list of one or more locations where an OMS Server resides. When this value is specified, it sets the *oms_servers* value inside the Universal Automation Center (UAG) Server configuration file. The UAG Server uses this value to determine which OMS Server it will connect to. The format of this value is **port@ipaddr**[;**port**1@**ipaddr**1[;...;**port**n@**ipaddr**n]], where **port** is the OMS Server's configured *oms_port* value and **ipaddr** is the IP address or host name of the OMS Server system. UAG Server will stop processing the list after it successfully connects to one of the OMS Servers listed.	-oms_servers *value*	UMIOMSSERVERS=*value*	oms_servers *value*	(none)
RUNBROKERASUSER	Controls whether the Universal Broker service executes as an Administrative user account or as **Local System**. • If **yes**, the install will perform the following steps: 1. The install will set the service's start-up account to **UBrokerService** (the default) or the user ID specified by the BROKERUID command line option. 2. If the Universal Broker service is installed and already is configured to run as a user account, the Broker's start-up account will not be changed (allows customized user accounts to be preserved during upgrades). • If **no**, the install will perform the following steps: 1. The install will set the service's start-up account to **Local System**. 2. If the Universal Broker service is installed and is currently configured to run as **LocalSystem**, the Universal Broker service's properties will not be changed.	-run_broker_as_user *opt*	UMIRUNBROKERASUSER=*opt*	run_broker_as_user *opt*	**yes**
SILENTMODE	When specified, no product installation dialogs are shown. If no other install parameters are specified, default values for all options are used. This option has no value. Only the parameter itself is necessary.	-silent	N/A	N/A	**no**
UAG_AUTOSTART	Specifies the behavior of the Universal Automation Center (UAG) Server component at Universal Broker startup. • If **yes** is specified, the AUTOMATICALLY_START UAG component definition option is set to **yes**, which instructs Universal Broker to start the UAG Server automatically when it is started. • If **no** (the default) is specified, or if this option is not included on the command line, the AUTOMATICALLY_START and RESTART UAG component definition options are set to **no**, and Universal Broker will not start the UAG Server automatically when it is started. However, the UAG Server may still be started manually any time after the Broker is started via the Universal Agent's Universal Control utility.	-uag_autostart *opt*	UMIUAGSRVAUTOSTART=*yes*	uag_autostart *opt*	**yes**
WORKING_FOLDER	Sets the Universal Agent working folder for this user mode installation. For user mode installs, this folder is shared by the following agent components: • Universal Command Server • Universal Control Server • Universal Data Mover Server • Universal Event Monitor Server If the folder contains spaces, you must use double (**"**) quotation marks around the path name.	-working_folder *dir*	UMIWORKINGFOLDER=*dir*	working_folder *dir*	**INSTALLDIR\UAHome**

Command Line Installation Examples

The following examples illustrate different ways that Universal Agent for Windows can be installed from the command line.

Graphical User Interface Install

To launch the graphical user interface for a Universal Agent for Windows User Mode Install via the command line, execute the following command:

```
sb-7.2.0.0-windows-x64-um.exe
```

Silent Install

To suppress the GUI and execute a silent Universal Agent for Windows User Mode Install, execute the following command:

```
sb-7.2.0.0-windows-x64-um.exe -silent
```

This will execute the install using default values for all options.

Silent Install with Universal Broker Service Options

The configure the Universal Broker and override the default installation directory from a silent Universal Agent for Windows User Mode Install, execute the following command:

```
sb-7.2.0.0-windows-x64-um.exe -silent -broker_dispname "Universal Broker (6800)" -broker_svcname UBroker6800 -
broker_port 6800 -installdir "c:\ua\6800"
```

Actual values used for each of the options shown above will depend on your specific requirements, but this command will:

- Install the Universal Agent to **`c:\ua\6800`**
- Configure the Universal Broker service with a display name of **Universal Broker (6800)**
- Configure the Universal Broker service to register messages sent to the Windows Application event log with a message source of **Universal Broker (6800)**
- Configure the Universal Broker service with an internal name of**UBroker6800**
- Configure the Universal Broker service to accept connections from Universal Agent components on port **6800**

This information will also be used to populate the **c:\ua\6800\ubrsvc.bat** script, which may be used to start, stop, restart, or remove the Universal Broker service.

Silent Install, with Universal Automation Center (UAG) Server Options

To configure the UAG Server and override the default installation directory from a silent Universal Agent for Windows User Mode Install, execute the following command:

```
sb-7.2.0.0-windows-x64-um.exe -silent -uag_autostart yes -ac_netname UAGENT-6800 -oms_servers 7878@omshost.com -
installdir "c:\ua\6800"
```

Actual values used for the some of the options above will depend on your specific requirements, but this command will:

- Install the Universal Agent to **`c:\ua\6800`**
- Configure the UAG Server component to start automatically at Universal Broker startup
- Connect to an OMS Server that resides on **omshost.com**, which is accepting connections over port **7878**
- Cause the UAG Server to register with a Universal Controller using the name **UAGENT-6800**

Removing a Universal Agent User Mode Install

Universal Agent for Windows User Mode Install Removal

Use the following two-step process to remove a Universal Agent for Windows User Mode Install:

Step 1	Open an Administrative command prompt and switch to the Universal Agent for Windows user mode installation directory.
	
	Execute the `ubrsvc` script and pass it the **-remove** option.
	
	For example, assuming the Agent was installed to `C:\Users\DAN~1.MOR\AppData\Local\Temp\universal`, issue the following commands
	
	`cd C:\Users\DAN~1.MOR\AppData\Local\Temp\universal`
	`ubrsvc -remove`
	
	The **-remove** option will stop the Universal Broker, delete it from the Windows Service Control Manager, and remove its message source registration with the Application event log viewer.

Step 2	After `ubrsvc` finishes, remove all the files and directories under the Universal Agent for Windows user mode installation location.

Universal Agent for Windows - User Mode Installation File Inventory Lists

- Universal Agent for Windows - File Inventory Lists
- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Connector for PeopleSoft
- Universal Connector for SAP
- Universal Control Manager
- Universal Control Server
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Copy
- Universal Encrypt
- Universal Event Log Dump
- Universal Message Translator
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- Universal Message Service (OMS)
- Universal Controller Command Line Interface (CLI)
- Universal Translate Tables (UTT)

Universal Agent for Windows - File Inventory Lists

The Universal Agent 7.2.x for Windows user mode installation package includes the files required for the following components / utilities:

- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Configuration Manager
- Universal Connector for PeopleSoft
- Universal Connector for SAP
- Universal Control Manager
- Universal Control Server
- Universal Copy
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Encrypt
- Universal Event Log Dump
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- Universal Message Service (OMS)
- Universal Controller Command Line Interface (CLI)

This page lists the files installed for a user mode installation and the Universal Agent for Windows component to which they belong. The file paths specified are relative to the root installation directory (for example, `%TEMP%\Universal`) that was specified during the installation.

Additionally, the following script files also are provided for a user mode installation:

File	Description
ubrsvc.bat	Manages the user mode-installed Universal Broker.
SB_ConfigSetup.bat	Configures the local environment for the user mode manager and utilities.

Universal Automation Center Agent

File	Description
UAGSrv\bin\uagsrv.exe	Universal Automation Center Agent Application program.
UAGSrv\bin\ops_copyfile.exe	Universal Automation Center Agent copy utility.
UAGSrv\bin\ops_scan.exe	Universal Automation Center Agent scan utility.
UAGSrv\bin\uftp.exe	Universal FTP Client program.
UAGSrv\samples\OPSWISE-MIB.txt	MIB file.
UAGSrv\samples\parser.txt	Universal FTP Client file describing the format of file listings received from FTP servers, based on operating system and/or file system.
UBroker\tmpl\uagcmp	Template file for the Universal Automation Center Agent component definition.
UBroker\tmpl\uagcfg	Template file for the Universal Automation Center Agent configuration.
nls\uagmceng.umf	English message catalog.
conf\uags.conf	Universal Automation Center Agent configuration file.
comp\uag	Universal Automation Center Agent component definition file.

Universal Broker

File	Description
UBroker\bin\ubroker.exe	Console application program.
UBroker\bin\ubrsvc.exe	Windows service program.
UBroker\bin\ubrdbrec.bat	Recovers Universal Agent database files.
UBroker\bin\xpsmsgnt.dll	Universal Broker file, used to interface with the Windows Application Event Log, containing IDs for the messages that Universal Broker may display.
UBroker\tmpl\aclcfg	Universal Access Control List (UACL) configuration template.
UBroker\tmpl\ubrcfg	Universal Broker configuration template.
nls\ussmceng.umc	English message catalog.
conf\uacl.conf	Universal Access Control List (ACL) configuration file.
conf\ubroker.conf	Universal Broker configuration file.

Universal Command Manager

File	Description
System32\bin\ucmd.exe	Universal Command Manager application program.
UBroker\tmpl\ucmcfg	Universal Command Manager configuration template.
nls\ucmmceng.umc	English message catalog.
conf\ucmd.conf	Universal Command Manager configuration file.

Universal Command Server

File	Description
UBroker\tmpl\ucscfg	Universal Command Server configuration template.

UBroker\tmpl\ucmcmp	Universal Command Server component definition template.
UCmdSrv\bin\ucmsrv.exe	Universal Command Server application program.
nls\ucmmceng.umc	English message catalog.
Sys32\bin\urc.exe	Universal Return Code utility.
comp\ucmd	Universal Command Server component definition file.
conf\ucmds.conf	Universal Command Server configuration file.

Universal Connector for PeopleSoft

File	Description
UBroker\tmpl\uppcfg	Universal Connector for PeopleSoft configuration template.
UPPS\bin\upps.exe	Universal Connector for PeopleSoft application program.
nls\uppmceng.umc	English message catalog.
conf\upps.conf	Universal Connector for PeopleSoft configuration file.

Universal Connector for SAP

File	Description
UBroker\tmpl\uspcfg	Universal Connector for SAP configuration template.
USap\bin\usap.exe	Universal Connector for SAP application program.
USap\samp\sapnwrfc.ini	SAP NW (NetWeaver) RFC interface configuration file in the sample file directory.
UBroker\tmpl\uspcfg	Universal Connector for SAP remote configuration template.
nls\uspmceng.umc	Universal Connector for SAP English message catalog.
nls*.stt	Spoollist translation table files.
conf\usap.conf	Universal Connector for SAP configuration file.

Universal Control Manager

File	Description
System32\bin\uctl.exe	Universal Control Manager application program.
UBroker\tmpl\uctcfg	Universal Control Manager configuration template.
nls\uctmceng.umc	English message catalog.
conf\uctl.conf	Universal Control Manager configuration file.

Universal Control Server

File	Description
UBroker\tmpl\utscfg	Universal Control Server configuration template.
UBroker\tmpl\uctcmp	Universal Control Server component definition template.
UCtlSrv\bin\uctsrv.exe	Universal Control Server application program.
nls\uctmceng.umc	English message catalog.

comp\uctl	Universal Control Server component definition file.
conf\uctls.conf	Universal Control Server configuration file.

Universal Data Mover Manager

File	Description
System32\bin\udm.exe	Universal Data Mover Manager application program.
UBroker\tmpl\udmcfg	Universal Data Mover Manager configuration template.
nls\udmmceng.umc	English message catalog.
conf\udm.conf	Universal Data Mover Manager configuration file.

Universal Data Mover Server

File	Description
UBroker\tmpl\udscfg	Universal Data Mover Server configuration template.
UBroker\tmpl\udmcmp	Universal Data Mover Server component definition template.
UdmSrv\bin\udmsrv.exe	Universal Data Mover Server application program.
nls\udmmceng.umc	English message catalog.
comp\udm	Universal Data Mover Server component definition file.
conf\udms.conf	Universal Data Mover Server configuration file.

Universal Event Monitor Manager

File	Description
System32\bin\uem.exe	Universal Event Monitor Manager application program.
UBroker\tmpl\uemcfg	Universal Event Monitor Manager configuration template.
nls\uemmceng.umc	English message catalog.
conf\uem.conf	Universal Event Monitor Manager configuration file.

Universal Event Monitor Server

File	Description
System32\bin\uemload.exe	Event definition and event handler database load utility.
UBroker\tmpl\uedcmp	Universal Event Monitor Demand-Driven Server component definition template.
UBroker\tmpl\uemcmp	Universal Event Monitor Event-Driven Server component definition template.
UBroker\tmpl\uescfg	Universal Event Monitor Server configuration template.
UemSrv\bin\uemsrv.exe	Universal Event Monitor Server application program.
nls\uemmceng.umc	English message catalog.
comp\uemd	Component definition file for the Universal Event Monitor Demand-Driven Server.
comp\uems	Component definition file for the Universal Event Monitor Event-Driven Server.
conf\uems.conf	Universal Event Monitor Server configuration file.

Universal Copy

File	Description
System32\bin\ucopy.exe	Universal Copy application program.

Universal Encrypt

File	Description
System32\bin\uencrypt.exe	Universal Encrypt application program.

Universal Event Log Dump

File	Description
System32\bin\ueld.exe	Universal Event Log Dump application program.
UBroker\tmpl\uelcfg	Universal Event Log Dump configuration template.
nls\uelmceng.umc	English message catalog.
conf\ueld.conf	Universal Event Log Dump configuration file.

Universal Message Translator

File	Description
System32\bin\umet.exe	Universal Message Translator application program.

Universal Products Install Merge

File	Description
UPIMerge\bin\upimerge.exe	Universal Products Install Merge application program.
UPIMerge\bin\upimerge.dll	Universal Products Install Merge exported function library.

Universal Query

File	Description
System32\bin\uquery.exe	Universal Query application program.
UBroker\tmpl\uqrcfg	Universal Query configuration template.
nls\uqrmceng.umc	English message catalog.
conf\uquery.conf	Universal Query configuration file.

Universal Spool Utilities

File	Description
USpool\bin\uslist.exe	Universal Spool List application program.
USpool\bin\uslrm.exe	Universal Spool remove application program.

USpool\bin\udb_archive.exe **USpool\bin\udb_dump.exe** **USpool\bin\udb_load.exe** **USpool\bin\udb_recover.exe** **USpool\bin\udb_stat.exe** **USpool\bin\udb_upgrade.exe** **USpool\bin\udb_verify.exe**	Miscellaneous spool file utilities. Used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support.

Universal Message Service (OMS)

File	Description
omssrv\bin\omssrv.exe	OMS Server application program.
omssrv\bin\omsadm.exe	OMS Administration utility.
UBroker\tmpl\omscfg	OMS Server configuration template.
UBroker\tmpl\omscmp	OMS Server component definition template.
nls\omsmceng.umc	OMS message catalog.
conf\omss.conf	OMS configuration file.
comp\oms	OMS component definition file.

Universal Controller Command Line Interface (CLI)

File	Description
OpsCli\bin\uagcmd.exe	CLI application program.
OpsCli\bin\ops-agent-status.bat	Displays Universal Agent status.
OpsCli\bin\ops-change-credentials-password.bat	Changes the runtime password for the specified Universal Controller credentials resource.
OpsCli\bin\ops-change-user-password.bat	Changes the password for the specified Universal Controller User account.
OpsCli\bin\ops-connector-status.bat	Lists the status of one or more Connectors.
OpsCli\bin\ops-export-bulk.bat	Performs a bulk export of all Controller database records.
OpsCli\bin\ops-export-trigger.bat	Performs an export of the specified triggers and any associated records.
OpsCli\bin\ops-import-bulk.bat	Imports Controller database records that were exported with ops-export-bulk.
OpsCli\bin\ops-import-trigger.bat	Imports triggers and associated records that were exported using ops-export-trigger.
OpsCli\bin\ops-manual-setcompleted.bat	Sets Manual task status to Success.
OpsCli\bin\ops-manual-setstarted.bat	Changes Manual task status from Action Required to Started.
OpsCli\bin\ops-resume-agent.bat	Allows a suspended Agent to submit tasks once again.
OpsCli\bin\ops-resume-agent-cluster.bat	Allows a suspended Agent Cluster to execute tasks once again.
OpsCli\bin\ops-resume-agent-cluster-membership.bat	Allows a specified Agent to rejoin specified Agent Cluster.
OpsCli\bin\ops-set-agent-cluster-task-execution-limit.bat	Sets the task execution limit for an Agent Cluster.
OpsCli\bin\ops-set-agent-task-execution-limit.bat	Sets the task execution limit for an Agent.
OpsCli\bin\ops-suspend-agent.bat	Temporarily prevents a specified Agent from submitting tasks.
OpsCli\bin\ops-suspend-agent-cluster.bat	Temporarily prevents the Agents in the specified Cluster from submitting tasks.
OpsCli\bin\ops-suspend-agent-cluster-membership.bat	Temporarily removes an Agent from cluster membership.
OpsCli\bin\ops-task-cancel.bat	Cancels a task.

OpsCli\bin\ops-task-forcefinish.bat	Force Finishes a task.
OpsCli\bin\ops-task-hold.bat	Places a task on hold.
OpsCli\bin\ops-task-launch.bat	Executes a task.
OpsCli\bin\ops-task-list.bat	Lists the specified tasks.
OpsCli\bin\ops-task-release.bat	Releases a held task.
OpsCli\bin\ops-task-rerun.bat	Re-executes the specified task.
OpsCli\bin\ops-task-setpriority.bat	Changes the execution priority of a Started task.
OpsCli\bin\ops-task-skip.bat	Skips the specified task instance.
OpsCli\bin\ops-task-status.bat	Displays the status of the task instance(s) associated with a task.
OpsCli\bin\ops-trigger-disable.bat	Disables the specified trigger(s).
OpsCli\bin\ops-trigger-enable.bat	Enables the specified trigger(s).
OpsCli\bin\ops-trigger-now.bat	Satisfies all conditions in the specified trigger and launches its associated tasks.
OpsCli\bin\ops-trigger-status.bat	Displays the status of the specified trigger(s).
OpsCli\bin\ops-update-resource-limit.bat	Sets the resource limit for a resource.
OpsCli\bin\ops-variable-list.bat	Displays the specified variable's value.
OpsCli\bin\ops-variable-set.bat	Sets the specified variable's value.

Universal Translate Tables (UTT)

File
nls\default.stt
nls\raw.stt
nls\iso8859-10.utt
nls\iso8859-13.utt
nls\iso8859-14.utt
nls\iso8859-15.utt
nls\iso8859-1.utt
nls\iso8859-2.utt
nls\iso8859-3.utt
nls\iso8859-4.utt
nls\iso8859-5.utt
nls\iso8859-6.utt
nls\iso8859-7.utt
nls\iso8859-8.utt
nls\iso8859-9.utt
nls\pc437.utt
nls\pc737.utt
nls\pc775.utt
nls\pc850.utt
nls\pc852.utt

nls\pc855.utt
nls\pc857.utt
nls\pc860.utt
nls\pc861.utt
nls\pc862.utt
nls\pc863.utt
nls\pc864.utt
nls\pc865.utt
nls\pc866.utt
nls\pc869.utt
nls\pc874.utt
nls\win1250.utt
nls\win1251.utt
nls\win1252.utt
nls\win1253.utt
nls\win1254.utt
nls\win1255.utt
nls\win1256.utt
nls\win1257.utt
nls\win1258.utt

Universal Enterprise Controller for Windows Installation

Currently, Windows runs Universal Enterprise Controller 6.3.0.0. These pages provide information for that version.

Overview

The following information is provided for the installation of Universal Enterprise Controller for Windows:

- Installation Package
- Installation Requirements
- Installation Procedures
- 64-Bit Windows Editions
- Database Configuration
- File Inventory Lists

(For licensing information, see Windows Installation - Licensing.)

Universal Enterprise Controller for Windows - Installation Package

- Components
- Component Compatibility

Components

The Universal Enterprise Controller (UEC) 6.3.x for Windows package includes a single component:

- Universal Enterprise Controller 6.3.x

Note

Installations of Universal Enterprise Controller v110-3 and earlier also included desktop application versions of the Universal Enterprise Controller Client Applications: I-Administrator, I-Activity Monitor, and I-Management Console.

As of v110-4, these client applications are contained in their own, separate installation package.

Component Compatibility

The following table identifies the compatibility of Universal Enterprise Controller 6.3.x for Windows with previous component / product versions.

Component	Compatibility
Universal Enterprise Controller 6.3.x	• Universal Broker 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 2.2.0. • Not compatible with previous versions of Universal Enterprise Controller Client Applications.

The component references pertain to all supported platforms for that version.

Universal Enterprise Controller for Windows - Installation Requirements

Windows Versions

To install the Universal Enterprise Controller, you must have one of the following versions of Windows:

- Windows Server 2003 SP1 and higher
- Windows XP SP3
- Windows Vista
- Windows 7
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

Additional Requirements

- An account with administrative privileges.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 35 megabytes of disk space.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Universal Enterprise Controller for Windows - Installation Procedures

Overview

The following procedures are provided for the installation and modification of Universal Enterprise Controller (UEC) for Windows:

- Installing UEC via the Graphical Interface
- Modifying a UEC Installation via the Graphical Interface
- Installing UEC via the Command Line
- Modifying a UEC Installation via the Command Line

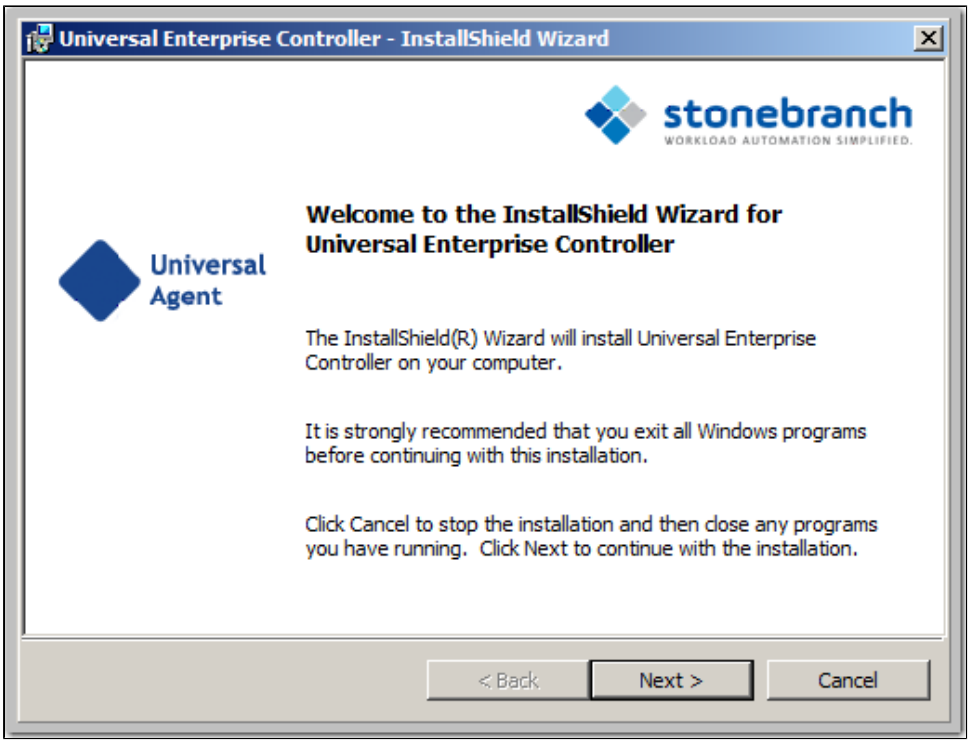
Note

Modifying a UEC installation refers to the adding / removing of UEC components, repairing a corrupted installation, or removing an installation. To change the installed version of UEC, see Upgrading Universal Agent and Applying Maintenance to Universal Agent.

Installing UEC via the Graphical Interface

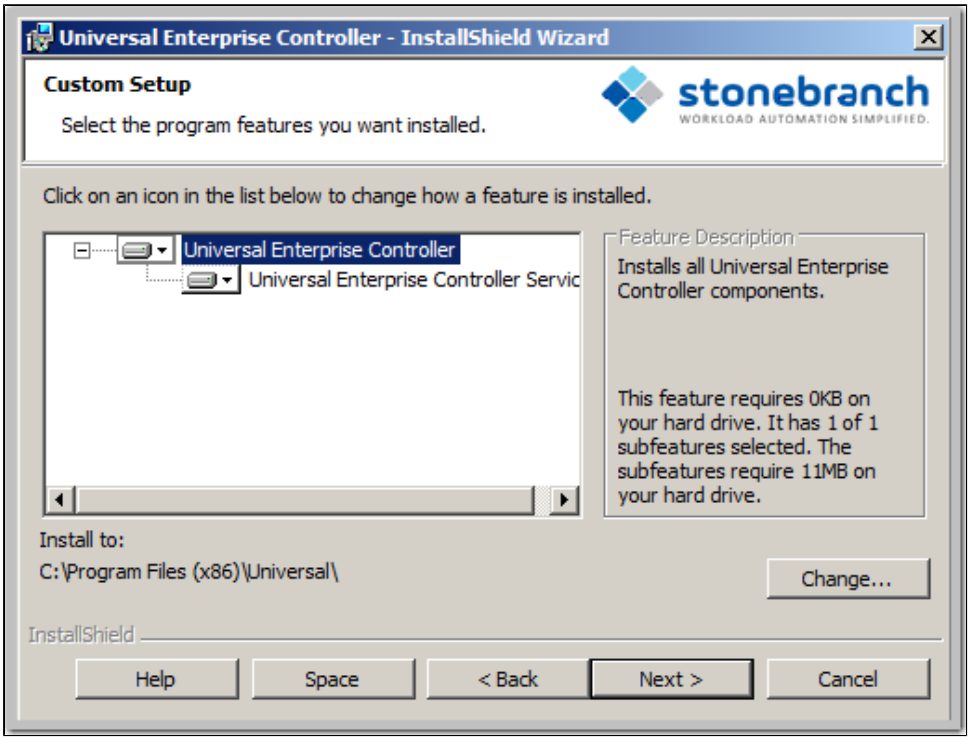
Installing Universal Enterprise Controller via the Windows Installer Graphical Interface

To install Universal Enterprise Controller using the Windows Installer graphical interface, perform the following steps:

Step 1	Download the Universal Enterprise Controller for Windows product distribution file, **sb-UECtlr-6.3.0.<level>-windows-i386.exe**, to your work station.
Step 2	Execute the distribution file to extract the files. Note If you already have extracted the files from the distribution file, but cancelled installation in order to separately install Windows Installer, you can simply double-click the extracted Universal Enterprise Controller installation file, **UECtlr.msi**, to begin the installation. Installing over a Remote Desktop Session Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user. This means that extraction may use an environment setting that is not available once the Remote Desktop session ends. For example, the distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. To ensure that extracted files and other required resources are accessible after the initial install, extract the files to a well-known location that is not likely to change between Remote Desktop sessions. Refer to the Microsoft documentation on the Remote Desktop feature for additional information. The installation starts after the files are extracted. It first will verify that your machine meets the minimum system requirements (see Universal Enterprise Controller for Windows - Installation Requirements). If the requirements are met, a Welcome dialog displays. 

Step 3

Click the **Next** button. A list of Universal Enterprise Controller components included in the installation package then displays. It is from this list that you can select which components to install.



For a new installation, a drive icon displays next to each item in the list, indicating that the component will be installed.

For an upgrade installation, either of the following icons displays next to an item:

- A drive icon indicates that the component is either:
  - New to the installation and will be installed.
  - Currently is installed and will be upgraded.
- An **X** icon indicates that the component is either:
  - Currently not installed (but previously was available).
  - Previously installed but removed.

A Stonebranch Tip

If the installation detects an existing Universal Enterprise Controller installation, currently installed components may be upgraded. (Currently, there is no way to specify that the state of a currently installed component remain unchanged.)
If a component is selected for installation, and the version of the installed component is below the version of the component being installed, the installed component will be replaced by the component being installed.
If a component is not selected for installation (that is, the **X** icon is selected), and it currently is installed, the new installation will remove that component.

Step 4

The previous figure shows that all Universal Enterprise Controller components will be installed in their respective directories under the **`C:\ProgramFiles\Universal`** directory.

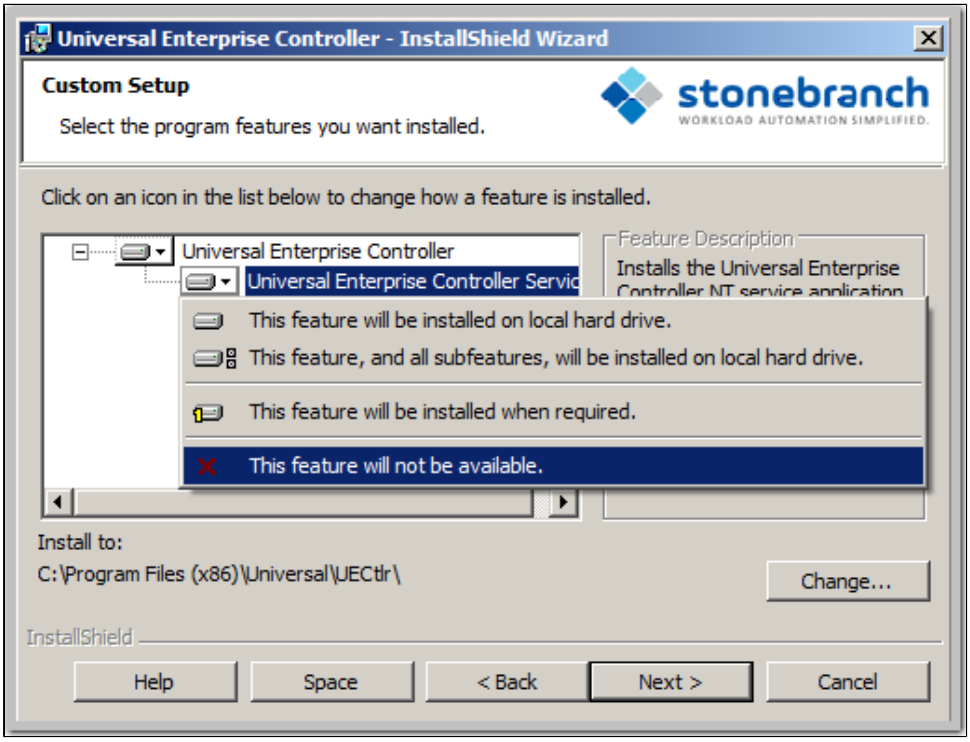
1. If you want to select a different location, click the **Change...** button.
2. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.

Step 5

If you do not want to install a component:

1. Click the drive icon next to that component name.
2. From the drop\-down list that displays, select the X icon to mark the component as one not to be installed.

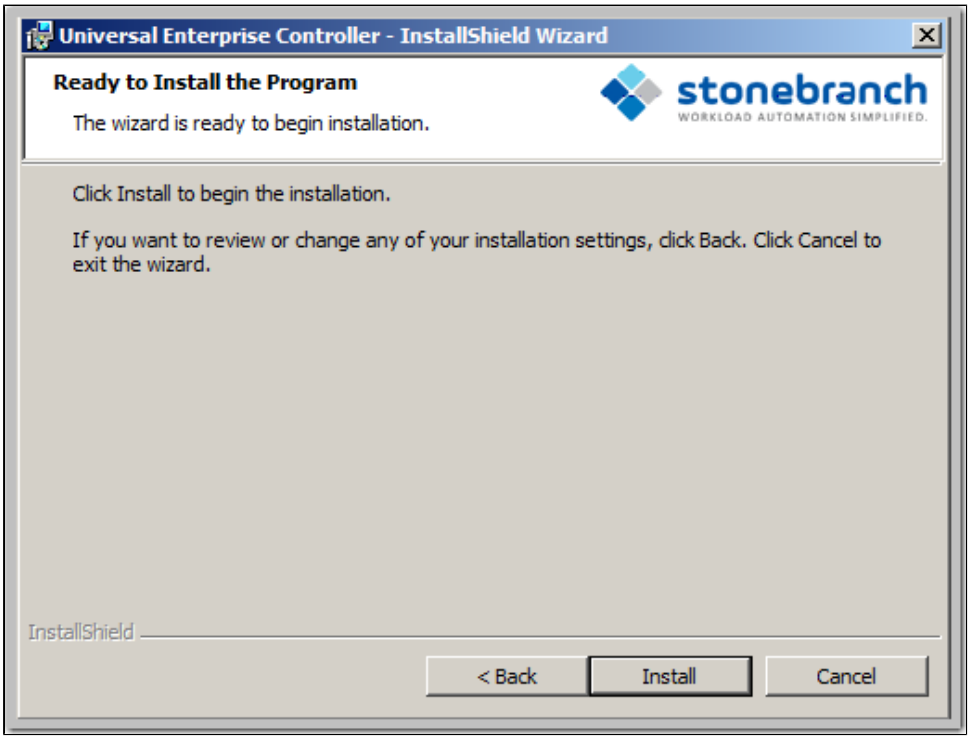
For example, the following figure indicates that the Universal Enterprise Controller Service has been selected to not be installed.



Step 6

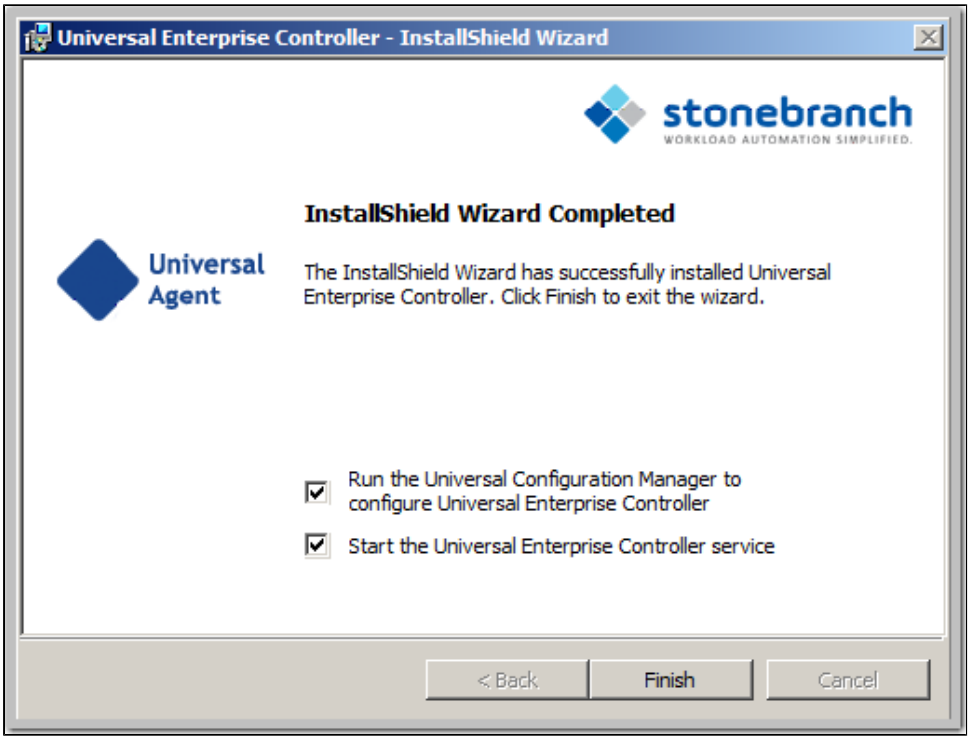
When you have selected the components (and their installation destinations) that you want to install, click the **Next>** button to continue the installation process.

When the installation is ready to begin, the Ready to Install dialog displays.



Click the **Install** button to begin the installation or click the **<Back** button to return to change information on any of the previous dialogs.

When the installation completes successfully, the Installation Complete dialog displays.



Step 7	If the Universal Enterprise Controller service was installed, the following options display on this dialog: • **Run the Universal Configuration Manager** • **Start the Universal Enterprise Controller service** Select both of these options to configure and/or start the Universal Enterprise Controller service.
Step 8	Click the **Finish** button to exit Windows Installation.

Modifying a UEC Installation via the Graphical Interface

- Modifying a Universal Enterprise Controller Installation via the Windows Installer Graphical Interface
- Adding or Removing Universal Enterprise Controller Components
- Repairing a Corrupted Universal Enterprise Controller Installation
- Removing a Universal Enterprise Controller Installation
    - Un-Installed Files

Modifying a Universal Enterprise Controller Installation via the Windows Installer Graphical Interface

After Universal Enterprise Controller is installed, Windows Installer can be run as many times as needed to modify the installation.

These installation modifications are:

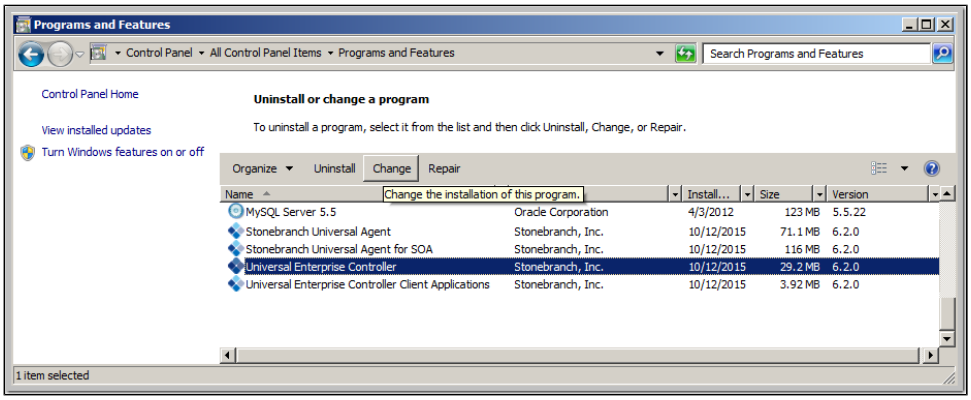
- Adding or Removing Universal Enterprise Controller Components
- Repairing a Corrupted Universal Enterprise Controller Installation
- Removing a Universal Enterprise Controller Installation

Adding or Removing Universal Enterprise Controller Components

To add or remove components from a Universal Enterprise Controller installation, perform the following steps:

Step 1

On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.



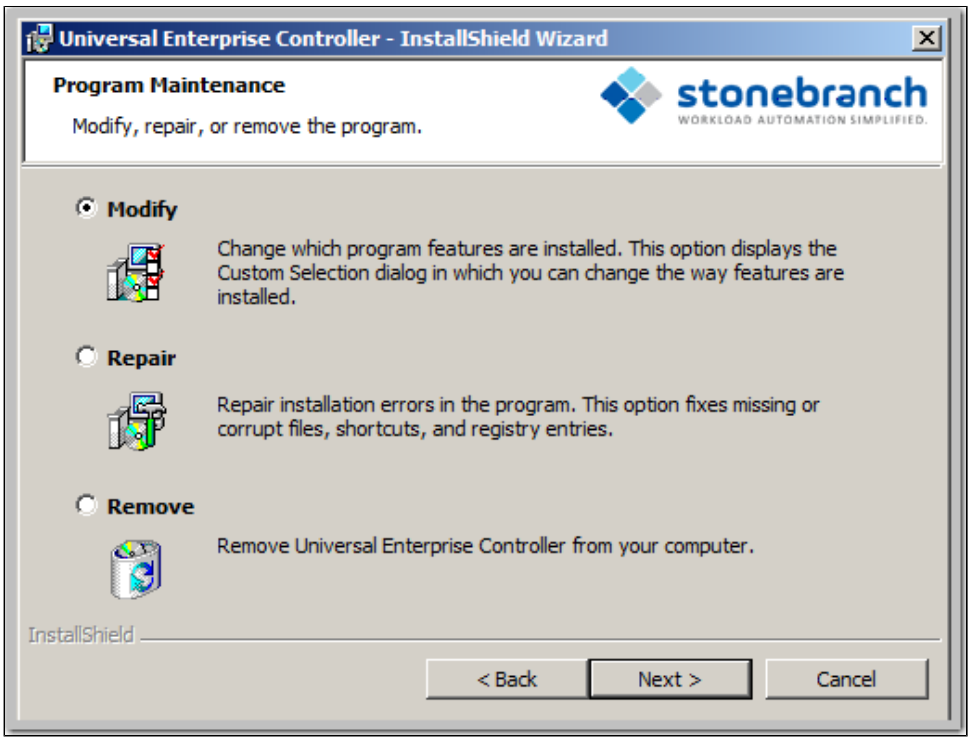
Windows Server 2003

If you are using Windows Server 2003, select **Add or Remove Programs** on the Windows Control Panel.

Step 2	From the list of installed programs, select **Universal Enterprise Controller** and click **Change** to start Windows Installer. The Welcome dialog displays.

Step 4

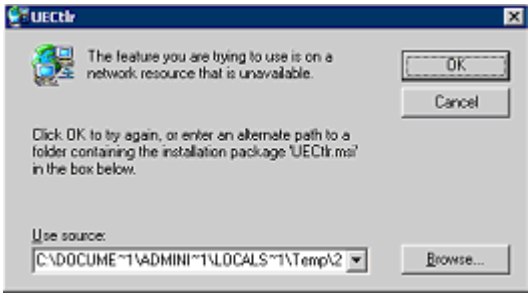
Click the **Next>** button. The Program Maintenance dialog displays.



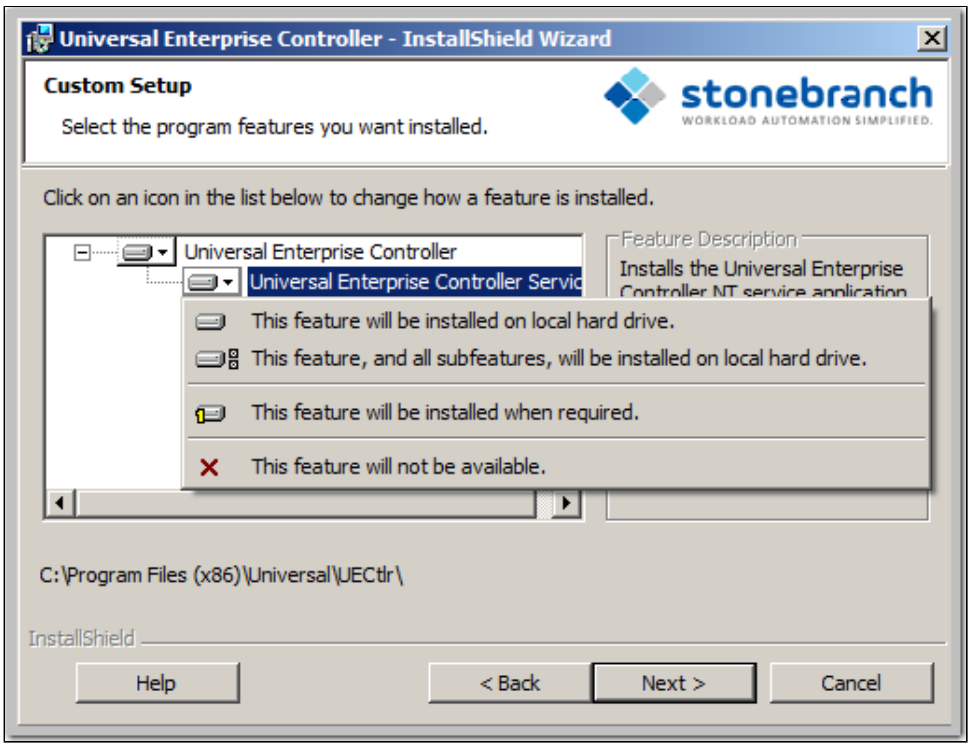
Installing over a Remote Desktop Session

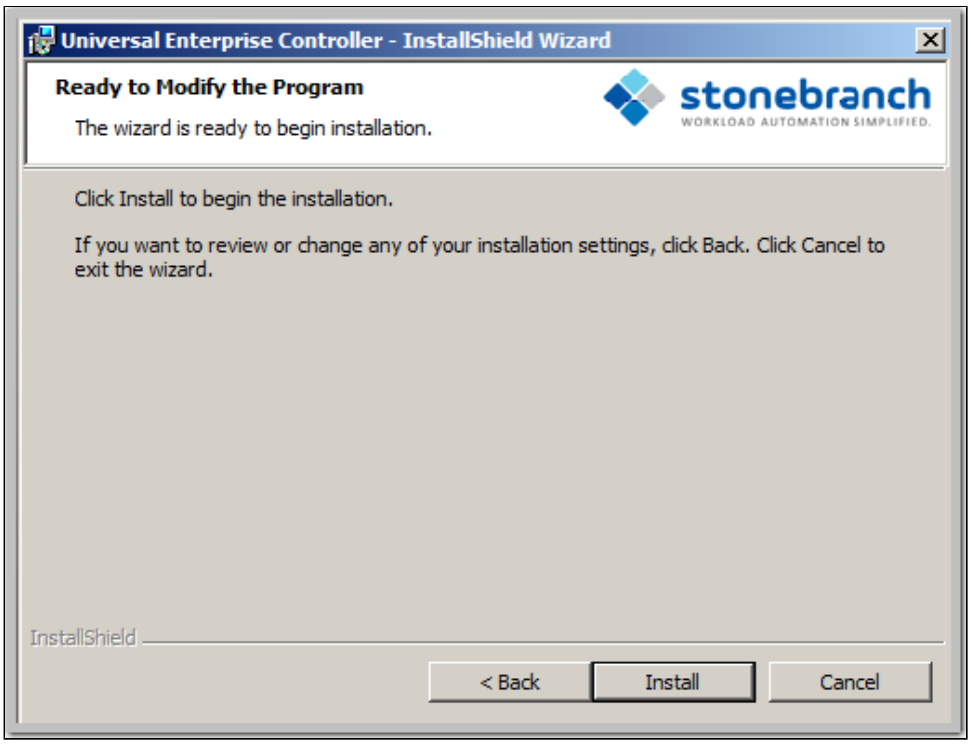
If Universal Enterprise Controller was installed via a Remote Desktop connection, the environment settings used during that session may no longer be available. Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user.

The distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. Consequently, any attempts to modify the installation may fail because the Windows Installer can't locate the installation's source files (a dialog similar to the one shown below may be displayed).



To resolve this issue, re-extract the distribution files to a location that is independent of a Remote Desktop environment and specify that location in the dialog above. Keep in mind, however, that the extracted files must come from the same distribution package used to do the initial install. If matching distribution files can't be found, Universal Enterprise Controller must be uninstalled and then reinstalled with the desired modifications.

Step 5	Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog.  Currently installed components are identified by a drive icon. Uninstalled components are identified by an **X** icon.
Step 6	To remove a currently installed component: 1. Click the drive icon next to that component. 2. Select the X icon from the drop-down list to mark the component for removal.
Step 7	To add an uninstalled component: 1. Click the X icon next to that component. 2. Select the drive icon from the drop-down list to mark the component for installation.

Step 8	Click the **Next>** button to display the Ready to Modify the Program dialog. 

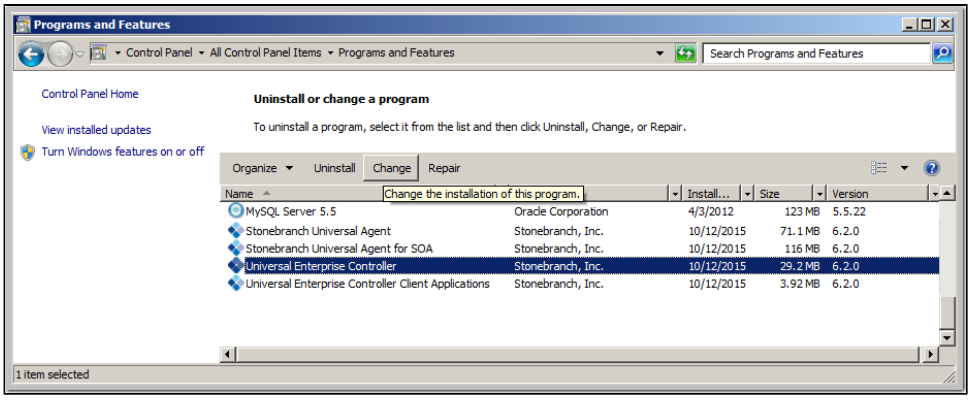
Step 9	Click the **Install** button to modify the installation.
	
	When the modifications are complete, the following actions will be taken:
	
	• Components marked with a drive icon will:
	  • Remain installed if they already are installed.
	  • Be installed if they are not already installed.
	• Components marked with an **X** will:
	  • Remain uninstalled if they are not currently installed
	  • Be removed if they currently are installed.

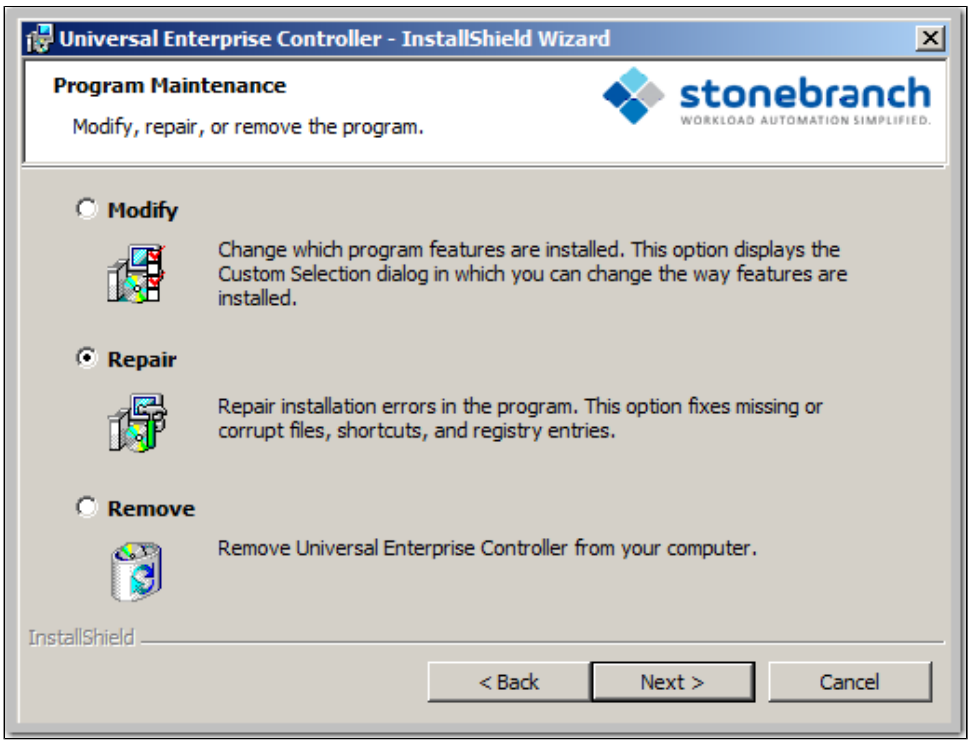
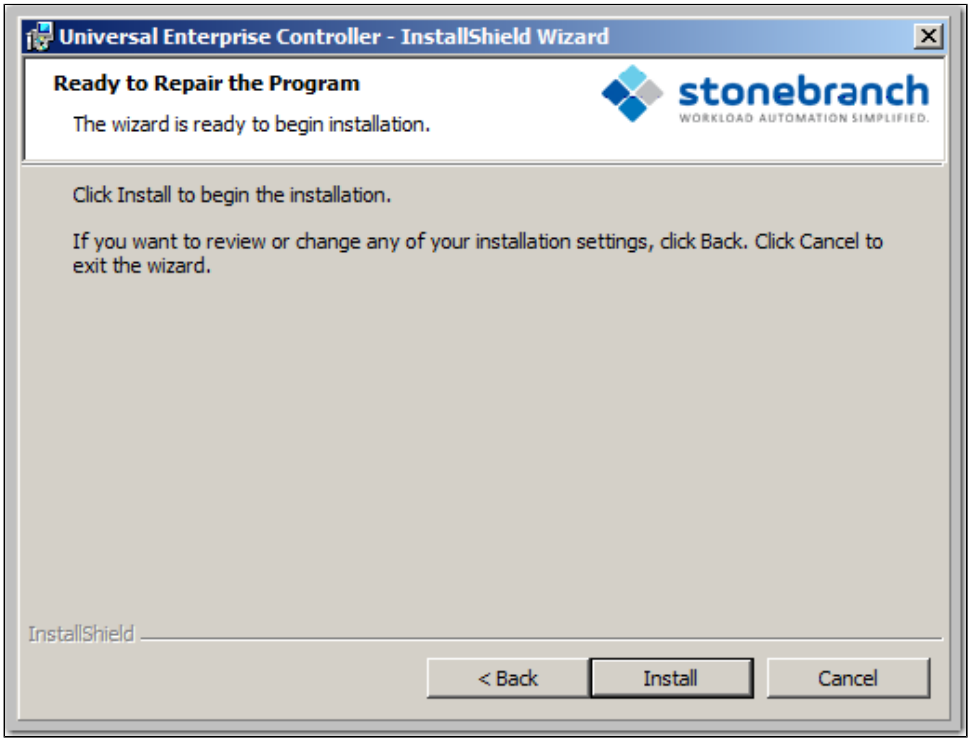
Repairing a Corrupted Universal Enterprise Controller Installation

Windows Installer has the ability to recover accidentally deleted application files, configuration files, or registry entries required by Universal Enterprise Controller. This repair feature will re-install the missing items, making a complete re-install unnecessary.

During a repair, any options stored in the Universal Enterprise Controller configuration file is preserved. If the UEC configuration file was deleted, the installation will create a new configuration file with default values.

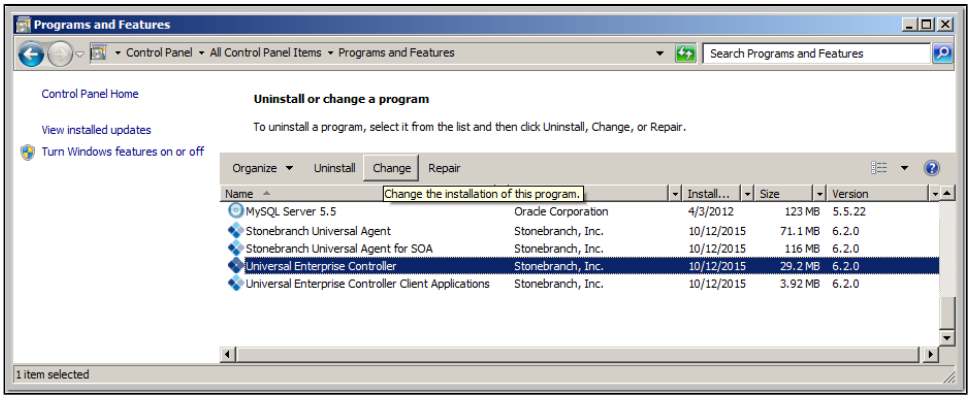
To repair an installation, perform the following steps:

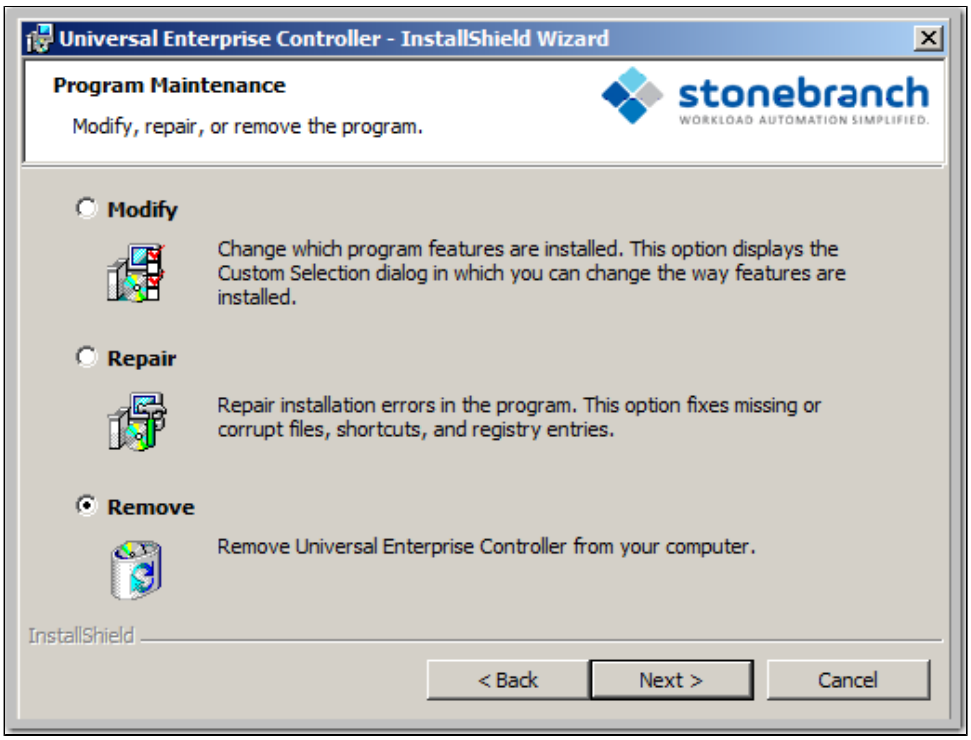
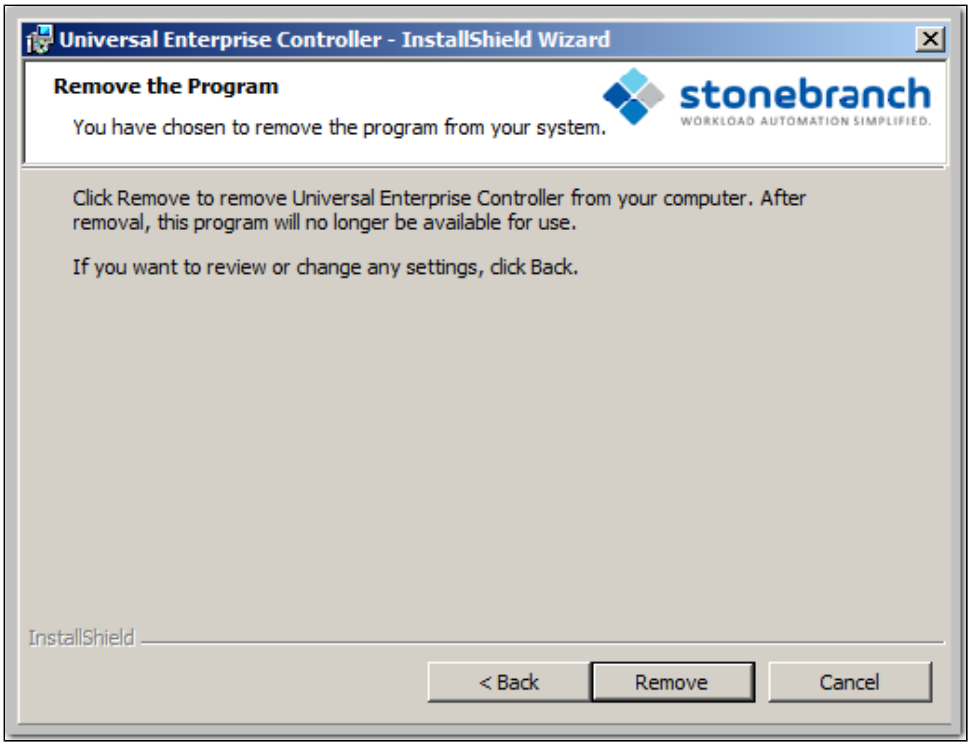
Step 1	On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.  pre-Vista versions of Windows If you are using an earlier version of Windows than Windows Vista, select **Add or Remove Programs** on the Windows Control Panel.
Step 2	From the list of installed programs, select **Universal Enterprise Controller** and click **Change** to start Windows Installer. The Welcome dialog displays.

Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 
Step 4	Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair the Program dialog. 
Step 5	Click the **Install** button to repair the installation.

Removing a Universal Enterprise Controller Installation

To uninstall a Universal Enterprise Controller installation, perform the following steps:

Step 1	On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.  pre-Vista versions of Windows If you are using an earlier version of Windows than Windows Vista, select **Add or Remove Programs** on the Windows Control Panel.
Step 2	From the list of installed programs, select **Universal Enterprise Controller** and click the **Change** button to start Windows Installer. The Welcome dialog displays.

Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 
Step 4	Click the **Remove** radio button, and then the **Next>** button, to display the Remove the Program dialog. 
Step 5	Click the **Remove** button to remove the installation.

Un-Installed Files

The un-install process will remove only those files created during the installation. Some files stored under the **.\Universal** install directory by Universal Enterprise Controller, such as trace files, may be left behind after the uninstall. In this situation, those files and/or directories may simply be deleted.

Before deleting the entire **.\Universal** directory, make sure that no other Stonebranch, Inc. products are installed there. (See Universal Enterprise Controller for Windows - File Inventory Lists for a list of files and directories installed with Universal Enterprise Controller.)

In addition to those files and directories created by the Universal Enterprise Controller installation, there may be some shared files left behind following an uninstall. These components will be removed when the last Stonebranch Inc. product that uses them is uninstalled.

Installing UEC via the Command Line

- Introduction
- Installing UEC
  - Command Line Syntax
  - Command Line Switches
  - Command Line Parameters
  - Command Line Installation Examples
- Detecting the Completion of Silent Installs

Introduction

This page describes how to install Universal Enterprise Controller (UEC) using the Windows Installer command line interface.

A command line installation is useful in situations where:

- Several Universal Enterprise Controller installations must be deployed.
- It is not practical or convenient to perform the graphical interface installation.
- It is necessary to generate an installation log file.

Installing UEC

Step 1	Download the Universal Enterprise Controller for Windows product distribution file, **sb-UECtlr-6.3.0.<level>-windows-i386.exe**, to your work station.
Step 2	Execute the distribution file from the command line, and include all appropriate command line switches and parameters. The installation process determines whether a Windows Installer update is needed. The process then extracts and saves a Windows installer package file (.msi) to this location. After all files (including the .msi) are extracted from the distribution file, the installation process verifies that your machine meets the minimum installation requirements. If the requirements are met, the installation begins.

Command Line Syntax

The following illustrates the command line syntax used to install UEC:

```
sb-UECtlr-6.3.0.<level>-windows-i386.exe [/v"command line parameters"] [/s] [/w] [/x]
```

In this syntax:

- **<level>** is the numeric package level.

The command line switches (/v, /s, /w, and /x) are processed directly by the distribution file to control behavior of the Windows Setup application.

The command line parameters are passed to the Windows Installer (**msiexec**) to control the extracted Windows installer package file (.msi) behavior during the install process.

Command Line Switches

The following table describes the command line switches available for a command line installation:

/v	Passes parameters to the Windows Installer (**msiexec**). The list of parameters must be enclosed in double (") quotation marks. See #Command Line Parameters for available parameters.
/s	Suppresses the initialization and extraction dialogs displayed before the product install Welcome dialog." If you are using the #/q command line parameter, use this switch additionally for a completely silent install.

/w	Instructs the Windows Setup application to wait until the installation completes. Use this switch when launching the installation from a script file. Without it, the Setup application may return immediately after launching Windows Installer.
/x	Uninstalls UEC.

Command Line Parameters

The following table describes the parameters that are available for a command line installation.

The parameters can be specified in any order, with the following exceptions:

- If the Repair (#/fom) or Remove (#/x) parameter is used, it must be specified ***before*** all other parameters.
- If the Silent install (#/q) and/or Log file (#/L) parameters are used, they can be specified in any order, but they must be specified ***after*** all other parameters.

These parameters are preceded by the #/v command line switch and must be enclosed in double (") quotation marks.

Parameter	Description	Default
/fom	Repairs a Universal Enterprise Controller installation. **om** (after the **/f**) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the **om** options must be used. **/fom** cannot be used with the **/x** (remove) parameter.	n/a
/x	Removes Universal Enterprise Controller. **/x** cannot be used with the **/fom** (repair) parameter.	n/a
INSTALLDIR =*installdir*	Sets the root installation directory to <**`installdir`**>. Each component will be installed under this directory. **INSTALLDIR** is required if you want to install UEC under a directory different from the one specified by the **PROGRAMFILES** environment variable (typically **`C:\Program Files\Universal`**). If the directory contains spaces, you must use double (**"**) quotation marks around the path name.	(none)
UECTLR={**yes**|**no**}	Specification for whether or not to install the UEC Service component during new installs, upgrades, or maintenance. - If **yes** is specified, UEC Service will be installed. - If **no** is specified, UEC Service will not be installed. If UEC Service already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UEC Service** is not required unless you want to change the current install state. **UEC Service** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **UEC Service** on the Custom Setup dialog when installing UEC via the graphical interface.	**yes**
/q	Suppresses the product installation dialogs. Use this parameter in addition to the #/s command line switch for a completely silent install. See #Command Line Switches, #Command Line Installation Examples, and #Detecting the Completion of Silent Installs for additional information regarding silent installs.	n/a
/L*v	Instructs the installation process to create an installation log file named <**`logfilepath`**> (full path name). If <**`logfilepath`**> contains spaces, you must enclose it with double (**"**) quotation marks. ***v** are flags used to specify the level of detail (**verbose**) contained in the log file. To reduce the amount of output generated, ***v** can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.	n/a

Command Line Installation Examples

The following examples illustrate different ways that UEC can be installed from the command line.

Graphical User Interface Install, All Components

To install all UEC components via the graphical user interface, issue the following command:

```
sb-UECtlr-6.3.0.0-windows-i386.exe
```

Graphical User Interface Install, All Components, with Log File

To install all UEC components using the Windows Installer graphical user interface and write a log file to **`C:\temp\install.log`** during the installation, issue the following command:

```
sb-UECtlr-6.3.0.0-windows-i386.exe /v"/l*v c:\temp\install.log"
```

Silent Install, Different Installation Directory

To install UEC under **D:\Universal**, which is a directory other than the one specified by the environment variable **PROGRAMFILES**, issue the following command:

```
sb-UECtlr-6.3.0.0-windows-i386.exe /s /v"/qn INSTALLDIR=D:\Universal"
```

Silent Install, All Components

To install all UEC components without using a graphical interface — that is, a silent install — issue the following command:

```
sb-UECtlr-6.3.0.0-windows-i386.exe /s /v"/qn"
```

Detecting the Completion of Silent Installs

If the #/q switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that may be used to detect when the Windows Installer process ends is to execute it using the system's **start** command. Using available command line switches, the **start** command can be used to initiate the Windows Installer process and then wait for it to finish. When the **start** command returns control to its calling process (for example, the command prompt), the process will have ended.

For example, from the command prompt, issue the following command to start the Universal Enterprise Controller installation and wait for it to finish.

```
start /b /wait sb-UECtlr-6.3.0.0-windows-i386.exe /w /s /v"/qn"
```

- The **/b** switch prevents the **start** command from opening a new window.
- The **/wait** parameter causes the **start** command to start the application, **sb-UECtlr-6.3.0.0-windows-i386.exe**, and then wait for it to finish.

The syntax above can also be used to execute the **start** command from within a script, such as a .**bat** file.

For more information on the **start** command, go to the Windows command prompt and enter: **start /?**.

Modifying a UEC Installation via the Command Line

- Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface
- Adding or Removing Universal Enterprise Controller Components
- Repairing a Corrupted Universal Enterprise Controller Installation
  - Silent Repair
  - Interactive Repair, with Log File
- Removing a Universal Enterprise Controller Installation
  - Silent Removal

Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface

This page describes how to modify a Universal Enterprise Controller installation via the Windows Installer command line.

After Universal Enterprise Controller is installed, Windows Installer can be run as many times as needed to modify the installation by:

- Adding or Removing Universal Enterprise Controller Components
- Repairing a Corrupted Universal Enterprise Controller Installation
- Removing a Universal Enterprise Controller Installation

(For a description of the parameters used in these procedures, see Windows Installer Command Line Parameters.)

Adding or Removing Universal Enterprise Controller Components

Currently, it only is possible to add or remove Universal Enterprise Controller components using the Windows Installer graphical interface (see Modifying a UEC Installation via the Graphical Interface).

Repairing a Corrupted Universal Enterprise Controller Installation

To recover accidentally deleted files or registry entries required by Universal Enterprise Controller using the Windows Installer command line interface, use the **/f** switch together with the **om** parameters.

These are the same repair options set internally by the graphical interface installation. They cause Windows Installer to reinstall files that are missing or older than the version contained in the Universal Enterprise Controller distribution file.

Silent Repair

To repair a Universal Enterprise Controller installation from the command line, issue the following command:

```
msiexec.exe /fom SetupPath\UECtlr.msi /q
```

Interactive Repair, with Log File

To repair a Universal Enterprise Controller installation using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to **`C:\Temp\repair.log`** while running the repair, issue the following command:

```
msiexec.exe /fom SetupPath\UECtlr.msi /L C:\Temp\repair.log
```

Removing a Universal Enterprise Controller Installation

To uninstall Universal Enterprise Controller using the Windows Installer command line interface, use the **/x** switch.

Silent Removal

To uninstall Universal Enterprise Controller without using the Windows Installer graphical interface, issue the following command:

```
msiexec.exe /x SetupPath\UECtlr.msi /q
```

Universal Enterprise Controller for Windows - 64-Bit Windows Editions

- Universal Enterprise Controller - Installing on 64-bit Windows Editions
- Applications Installed in the Windows System Folder
  - Example 1
  - Example 2

Universal Enterprise Controller - Installing on 64-bit Windows Editions

All Universal Agent components have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

This page describes some modifications that may need to be made to the default installation options to ensure that the installed Universal Enterprise Controller components function correctly.

Applications Installed in the Windows System Folder

The Universal Enterprise Controller package installs several command-line applications in the Windows system folder. The default system folder for 32-bit applications installed on 64-bit Windows editions is the **`%SystemRoot%\SysWOW64`** directory (for example, **`C:\Windows\SysWOW64`**).

The following table identifies the affected Universal Agent applications.

File Name	Description
ucert.exe	Universal Certificate
uecload.exe	UECLoad
uencrypt.exe	Universal Encrypt

These applications can be executed using either the:

- 32-bit command shell (**%SystemRoot%\SysWOW64\cmd.exe**)
- Default 64-bit command shell (**%SystemRoot%\System32\cmd.exe**).

By default, the **`%SystemRoot%\SysWOW64`** directory is not part of the system path. Therefore, to execute **uecload.exe** using the 32-bit command shell, either:

- Directory must be added to the **PATH** environment variable.
- Complete path to the application and/or the 32-bit command shell must be specified.

Example 1

To execute UECLoad in the default 64-bit command shell, issue the following command:

```
%SystemRoot%\SysWOW64\uecload
```

Example 2

To execute UECLoad within the 32-bit command shell, issue the following command:

```
%SystemRoot%\SysWOW64\cmd.exe /C %SystemRoot%\SysWOW64\uecload
```

Universal Enterprise Controller for Windows - Database Configuration

Universal Enterprise Controller - Database Configuration

Berkeley DB uses a temporary cache in memory to manage its databases. If this cache becomes sufficiently large, it must be written to disk.

Berkeley DB has a default location for storing temporary cache files, but if UEC cannot access that location, or there is no space to write these files in the default location, the following error can occur in UEC, and UEC shuts down:

UNV4301D Database error: 'temporary: write failed for page XXXXX'

To work around this issue, perform the following steps to write the temporary cache files to the UEC database directory:

Step 1	Inside the UEC database directory, create a text file named **DB_CONFIG**.
Step 2	Inside the **DB_CONFIG** file, add the following string: **`set_tmp_dir *dbpath*`** (**`dbpath`** is the path to the location in which the database files reside.)
Step 3	Start / restart UEC.

Universal Enterprise Controller for Windows - File Inventory Lists

- Universal Enterprise Controller - File Inventory Lists
- Universal Enterprise Controller
- Universal Products Install Merge Utility
- System Files

Universal Enterprise Controller - File Inventory Lists

The Universal Enterprise Controller installation includes the files required for the following components / utilities:

- Universal Enterprise Controller
- Universal Configuration Manager
- Universal Products Install Merge

The Universal Configuration Manager is installed whenever Universal Enterprise Controller is installed. It is used to maintain the configuration options for the Universal Enterprise Controller service. If any of the components already are installed, Windows Installer will upgrade them to the latest version.

The files installed with each Universal Enterprise Controller component / utility are listed in the following tables. The file paths specified are relative to the root installation directory (for example, **`C:\Program Files\Universal`**) that was specified during the installation.

Items shown with a path of System32 are installed in the 32-bit system folder. The actual name of this directory depends on the Windows version:

- For all supported 32-bit Windows editions, the path is **`\Windows\System32`**.
- For all supported 64-bit Windows editions, the path is **`\Windows\SysWow64`**.

Universal Enterprise Controller

File	Description
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\README.TXT	Information about the contents of the **.\nls** directory.
nls\uecmceng.umc	English message catalog.
UCfgMgr\bin\ucfgmgr.cpl	Universal Configuration Manager control panel application.
UCfgMgr\bin\ucfgmgr.hlp	Universal Configuration Manager help file.
UECtlr\bin\acl.xml	Used to store Access Control List entries for Universal Enterprise Controller.
UECtlr\bin\brokers.xml	Used to store a list of Universal Brokers that will be monitored by the Universal Enterprise Controller.
UECtlr\bin\groups.xml	Used to store defined groups of Universal Brokers.
UECtlr\bin\librfc32.dll	Support file for Universal Enterprise Controller.
UECtlr\bin\ueccfg.dll	Used by Universal Configuration Manager to manage Universal Enterprise Controller configuration options.
UECtlr\bin\ueccfg.hlp	Universal Enterprise Controller configuration help file.
UECtlr\bin\uecdbrec.bat	Recovers a Universal Enterprise Controller database, specified from the command line. The default database is **uec_evm.db**.
UECtlr\bin\uecmsgnt.dll	Used to write error messages to the Windows Application event log.

UECtlr\bin\uectlrs.exe	Universal Enterprise Controller installation file.
UECtlr\bin\users.xml	Used to store a list of user accounts authorized to use Universal Enterprise Controller and its associated Client Applications: I-Administrator, I-Activity Monitor, and I-Management Console.
UPIMerge\bin\upimerge.exe	Command line interface to configuration file merge routines).
USpool\bin\udb_dump.exe	UEC database utility; to be used only upon request of Stonebranch, Inc. Customer Support.
USpool\bin\udb_load.exe	UEC database utility; to be used only upon request of Stonebranch, Inc. Customer Support.
Universal\conf\uec.conf[1]	Universal Enterprise Controller configuration file.
ucert.exe[2]	Universal Certificate utility, used to generate X.509 Certificates.
uecload.exe[2]	Universal Enterprise Controller load utility.
uencrypt.exe[2]	Universal Encrypt utility, used to encrypt sensitive Universal application command line options.

[1] This file is installed under **`%ALLUSERSPROFILE%\ Application Data`**, which, by default, resolves to:

- **`C:\Documents and Settings\All Users\Application Data`** on 2003 and XP.
- **`C:\ProgramData`** on Vista and Windows 2008 Server.

[2] This file is installed under environment variable **`%SystemRoot%`**, which, by default, resolves to:

- **`C:\Windows`** on all platforms.
  - On x86 platforms, these files are installed under **`%SystemRoot%\system32`**.
  - On x64 platforms, these files are installed under **`%SystemRoot%\SysWow64`**.

Universal Products Install Merge Utility

File	Description
UPIMerge\bin\upimerge.exe	Application program, always installed. Provides command line access to the same functionality used by the Universal Agent installation to merge options from a new configuration file into an existing file.

System Files

The following files will be installed only if they are newer than the existing files.

The directories shown in this table are relative to the **`%SYSTEMROOT%`** directory, where **`%SYSTEMROOT%`** is an environment variable that resolves to **`C:\Windows`** on all Windows platforms.

File	Description
System32\asycfilt.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\comcat.dll	Version 4.71.1460.1 of the Microsoft Component Category Manager library.
Microsoft C-Runtimev8.0.50727.762[1]	Version 8.0.50727.762 of the Microsoft C runtime side-by-side assembly.
System32\msiexec.exe	Version 3.1.4000.1823 of the Microsoft Windows Installer (see Windows Installer for more information).
System32\oleaut32.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\olepro32.dll	Version 5.0.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\psapi.dll	Version 4.0.1371.1 of the Microsoft process status library.
System32\stdole2.tlb	Version 2.40.4275.1. This file is one of the components of the Microsoft OLE library.

[1] The Microsoft C-Runtime distribution consists of several files, which are subject to change. Refer to Microsoft documentation for a complete list of files delivered with the specified runtime version.

UEC Client Applications Installation

Currently, Windows runs UEC Client Applications 6.3.0.0. These pages provide information for that version.

UEC Client Applications

The following information is provided for the installation of UEC Client Applications:

- Installation Package
- Installation Requirements
- Installation Procedures
- 64-Bit Windows Editions
- File Inventory Lists

UEC Client Applications - Installation Package

Components

The Universal Enterprise Controller (UEC) 6.3.x Client Applications for Windows package includes the desktop application versions of the following components:

- I-Administrator 6.3.x
- I-Activity Monitor 6.3.x
- I-Management Console 6.3.x

Component Compatibility

The following table identifies the compatibility of Universal Enterprise Controller Client Applications 6.3.x with previous component / product versions.

Component	Compatibility
Universal Enterprise Controller Client Applications 6.3.x	Not compatible with previous versions of Universal Enterprise Controller for Windows.

The component references pertain to all supported platforms for that version.

UEC Client Applications - Installation Requirements

- Windows Versions
- Additional Requirements
- Platform Requirements
- Java Runtime Environment

Windows Versions

To install the UEC Client Applications, you must have one of the following versions of Windows:

- Windows Server 2003 SP1 and higher
- Windows Server 2003 R2
- Windows XP SP3
- Windows Vista
- Windows 7
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

Note

Itanium not supported for Windows Server 2003/2003 R2 and Windows Server 2008/2008 R2.

Additional Requirements

- For a per-machine install, Administrator access is required. For a per-user install, any account with the authority to install software can be used.
- The following conditions must be met before the UEC Client Applications can be successfully installed:
  - Account used for the installation must have write access to the desired destination folder.
  - No system policies (defined in Windows by your system administrator) may be in place that prohibit non-Administrative accounts from installing software.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 5 megabytes of disk space. This value does not include space required for the Java Runtime Environment (JRE). See the JRE installation requirements, available from Sun, for more information.
- Sun Java Runtime Environment (JRE 1.5 or above).

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Java Runtime Environment

To run the UEC Client Applications, you also must have the Java Runtime Environment (JRE) version 1.5 or above. You can download the latest JRE installation package directly from Sun's website, www.java.com.

A Stonebranch Tip

Changes in Sun's versioning scheme for Java has created some confusion.

With version 1.2 of the Java environment, Sun began referring to Java as Java 2. The formal name was actually Java 2 with SDK 1.2.

As of January 2008, version 1.6 is latest version of the Java environment - a newer version than the original Java 2.

UEC Client Applications - Installation Procedures

UEC Client Applications - Installation Procedures

The following procedures are provided for the installation and modification of UEC Client Applications:

- Installing UEC Client Applications via the Graphical Interface
- Modifying a UEC Client Applications Installation via the Graphical Interface
- Installing UEC Client Applications via the Command Line
- Modifying a UEC Client Applications Installation via the Command Line

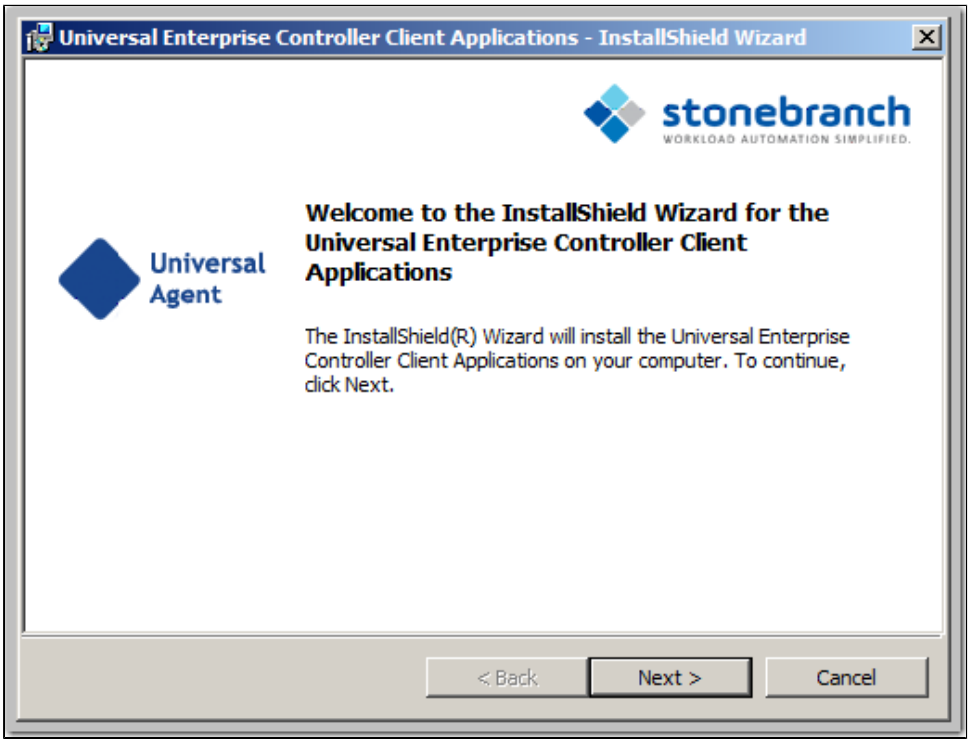
Note

Modifying a UEC Client Applications installation refers to the adding / removing of UEC Client Applications components, repairing a corrupted installation, or removing an installation. To change the installed version of UEC Client Applications, see Upgrading Universal Agent and Applying Maintenance to Universal Agent.

Installing UEC Client Applications via the Graphical Interface

Installing UEC Client Applications via the Windows Installer Graphical Interface

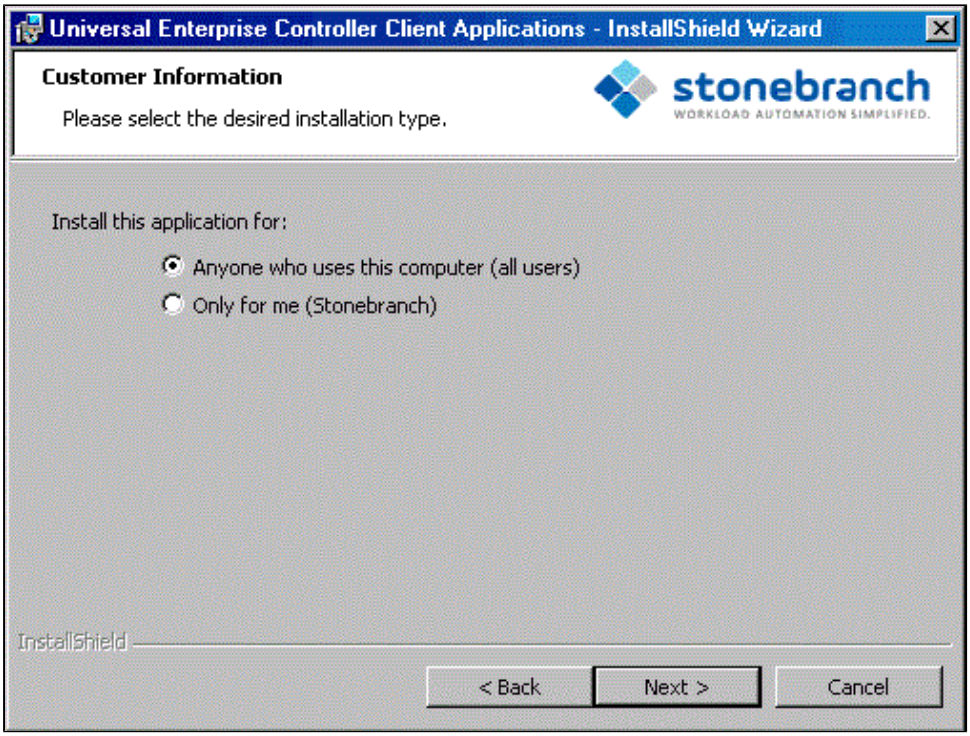
To install UEC Client Applications for Windows using the Windows Installer graphical interface, perform the following steps:

Step 1	Download the UEC Client Applications for Windows product distribution file, **`sb-UEClient-6.3.0.<level>-windows-i386.exe`**, to your work station.
Step 2	Execute the distribution file to extract the files. Note If you already have extracted the files from the distribution file, but cancelled installation in order to separately install Windows Installer (see Windows Installer), you can simply double-click the extracted Client Applications installation file, **`UEClient.msi`**, to begin the installation. Installing over a Remote Desktop Session Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user. This means extraction may use an environment setting that is not available once the Remote Desktop session ends. For example, the distribution file's default extraction location is based on the TEMP environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. To ensure that extracted files and other required resources are accessible after the initial install, extract the files to a well-known location that is not likely to change between Remote Desktop sessions. Refer to the Microsoft documentation on the Remote Desktop feature for additional information. The installation automatically will begin after the files are extracted. It first will verify that your machine meets the minimum system requirements (see UEC Client Applications - Installation Requirements). If the requirements are met, a Welcome dialog displays. 

Step 3

Click the **Next** button.

- If the installation is being performed by a user account that is a member of the Administrators group, the Customer Information dialog, below, displays.
- If the installation is being performed by a user account that is NOT a member of the Administrators group, the Custom Setup dialog displays (see Step 5).



Step 4

Click the appropriate radio button on this dialog:

- If you want to perform an installation for all users with access to a given workstation, click **Anyone who uses this computer**. This is referred to as a *per-machine* installation. When this type of installation is performed, shortcuts added to the **Start** menu, and any configuration settings stored in the Windows registry, are placed in a location accessible to anyone who logs on to that particular machine. This type of installation also creates an entry for UEC Client Applications in the **Add or Remove Programs** list, accessible via the Windows Control Panel. However, only accounts with administrative privileges will be able to:
  - Modify the installation (see Modifying a UEC Client Applications Installation via the Graphical Interface)
  - Uninstall the product (see Removing a UEC Client Applications Installation).

- If you want to perform an installation for the user identified on this dialog, click **Only for me**. (This is the account performing the installation.) This is referred to as a *per-user* installation. When this type of installation is performed, the Start menu shortcuts, the configuration options stored in the Windows registry, and the application itself (depending on where it is installed) will be accessible only by this user. This type of installation also creates an entry for UEC Client Applications in the **Add or Remove Programs** list, accessible via the Windows Control Panel. However, it will be visible only by this user.

  In this case, the user also will be able to:
  - Modify the installation (see Modifying a UEC Client Applications Installation via the Graphical Interface).
  - Uninstall the product (see Removing a UEC Client Applications Installation).For the UEC Client Applications to be available to another user on this machine, that user also must perform a _per\-user_ installation.

Note

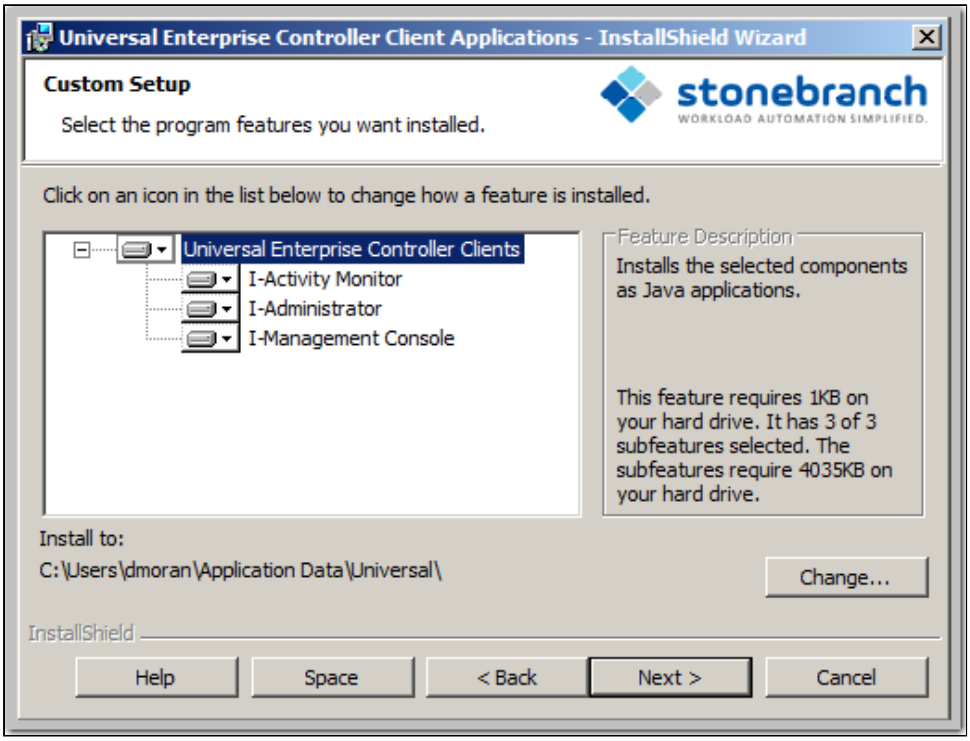
It is possible for a *per-machine* and a *per-user* installations to be performed on the same machine, although there is no benefit in doing this. In that situation, when any user that has done a *per-user* installation is logged on, that installation takes precedence. If the user removes the Client Applications, the *per-machine* installation will remain in effect. Conversely, if the *per-machine* installation is removed, users that have performed a *per-user* installation will still have access to Client Applications.

A Stonebranch Tip

For Non-Administrative Users:
Because *per-machine* installations require access to certain system resources to which most non-Administrative accounts do not have access, all installations performed by non-Administrative users are *per-user* installations.

Step 5

Click the **Next>** button. A list of UEC Client Applications components included in the installation package then displays. It is from this list that you can select which components to install.



For a new installation, a drive icon displays next to each item in the list, indicating that the component will be installed.
For an upgrade installation, either of the following icons displays next to an item:

- A drive icon indicates that the component is either:
  - New to the installation and will be installed.
  - Currently is installed and will be upgraded.
- An *X* icon indicates that the component is either:
  - Currently not installed (but previously was available).
  - Previously installed but removed.

A Stonebranch Tip

The directory identified in the figure above is typical for a new, *per-machine* installation. Depending on the type of installation being performed, the directory may be different.

For a new, *per-user* installation, the dialog will identify a directory located within the user's profile directory (for example, **`C:\Documentsand Settings\username`**).

For installation upgrades, whether *per-machine* or *per-user*, the UEC Client Applications' current location is displayed.

Step 6

The figure above shows that all UEC Client Applications will be installed in their respective directories under the **`C:\ProgramFiles\Universal`** directory.

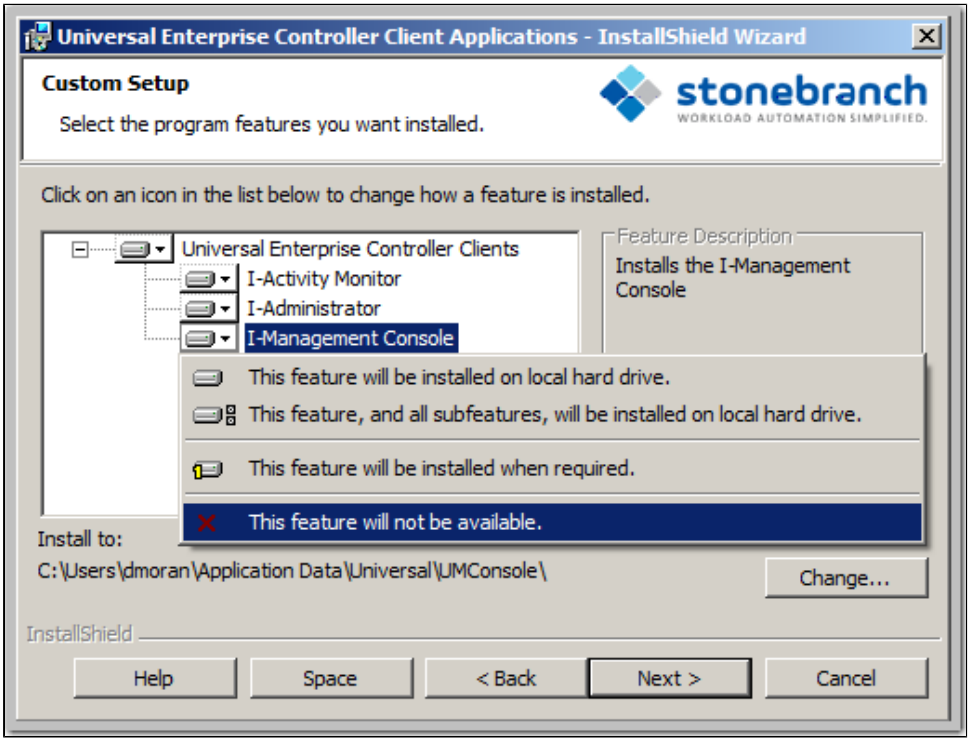
1. If you want to select a different location, click the **Change...** button.
2. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.

Step 7

If you do not want to install a component:

1. Click the drive icon next to that component name.
2. From the drop\-down list that displays, select the X icon to mark the component as one not to be installed.

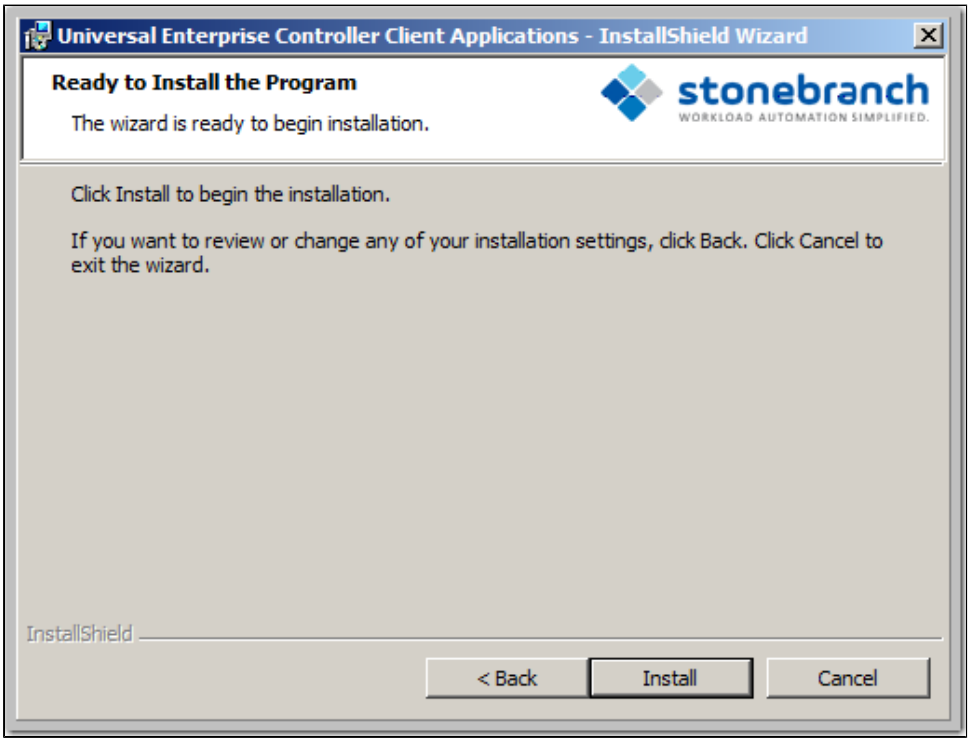
For example, the figure below indicates that I-Management Console has been selected to not be installed.



Step 8

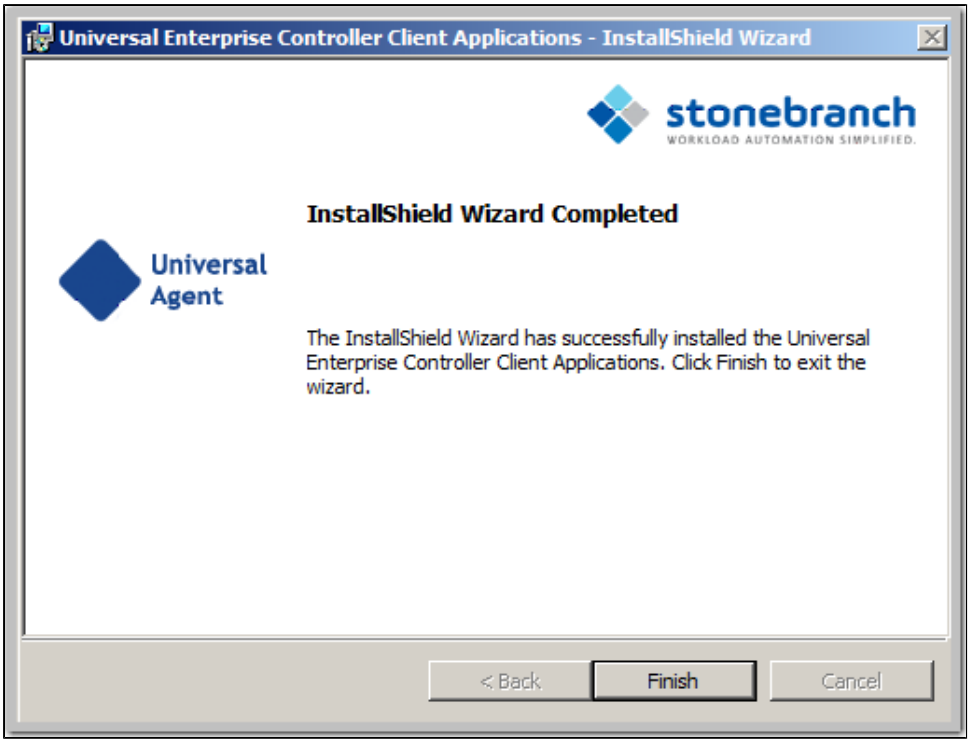
When you have selected the components (and their installation destinations) that you want to install, click the **Next** button to continue the installation process.

When the installation is ready to begin, the Ready to Install dialog displays.



Click the **Install** button to begin the installation or click the **<Back** button to return to change information on any of the previous dialogs.

When the installation completes successfully, the Installation Complete dialog displays.



Note

During the install, you may have been notified that no Java Runtime Environment (JRE) was detected on your system. The UEC Client Applications require Java Runtime Environment (JRE) Version 1.5 or greater (see UEC Client Applications - Installation Requirements).

Step 9	Click the **Finish** button to exit Windows Installation.

Modifying a UEC Client Applications Installation via the Graphical Interface

- Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface
- Adding or Removing UEC Client Applications Components
- Repairing a Corrupted UEC Client Applications Installation
- Removing a UEC Client Applications Installation
  - Un-Installed Files

Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface

After the UEC Client Applications are installed, Windows Installer can be run as many times as needed to modify the installation.

These installation modifications are:

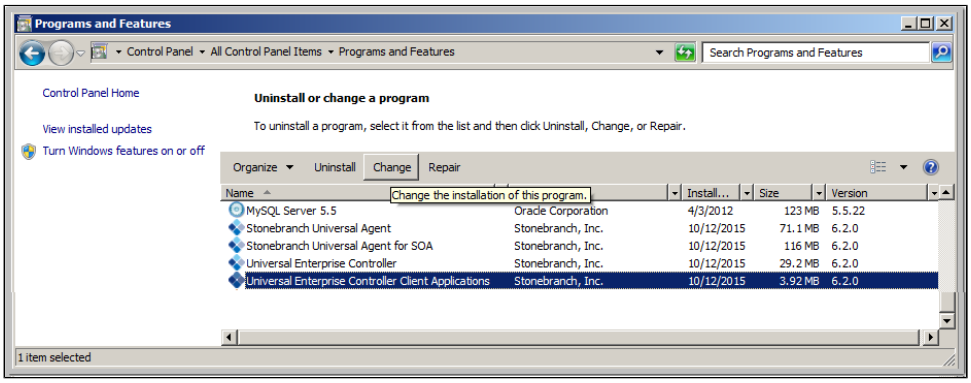
- Adding or Removing UEC Client Applications Components
- Repairing a Corrupted UEC Client Applications Installation
- Removing a UEC Client Applications Installation

Adding or Removing UEC Client Applications Components

To add or remove components from a UEC Client Applications installation, perform the following steps:

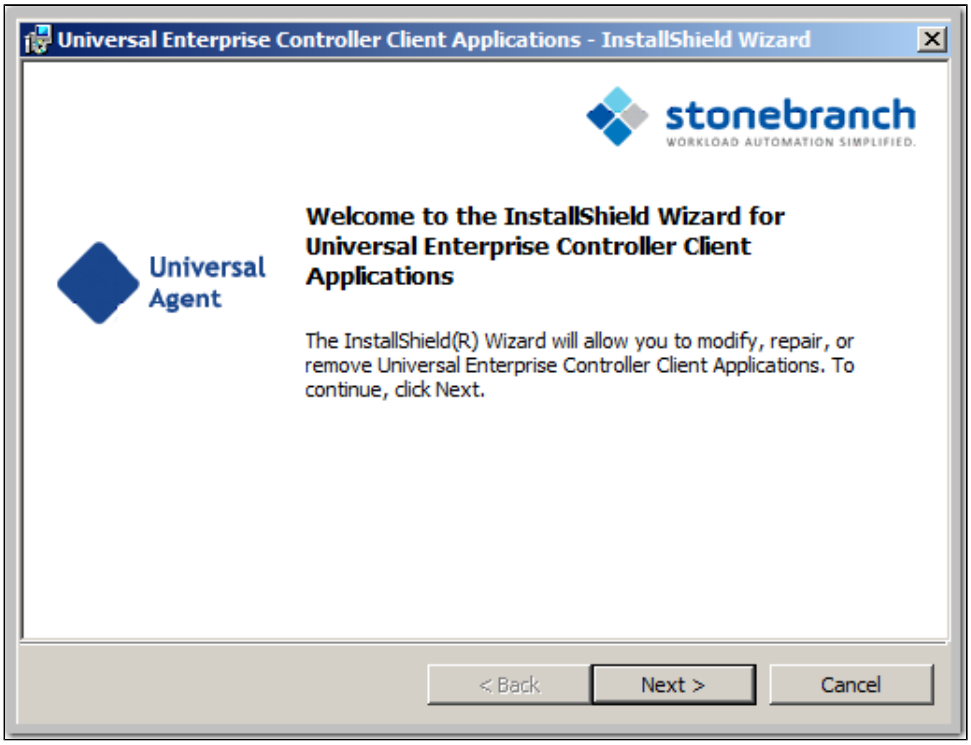
Step 1

On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.



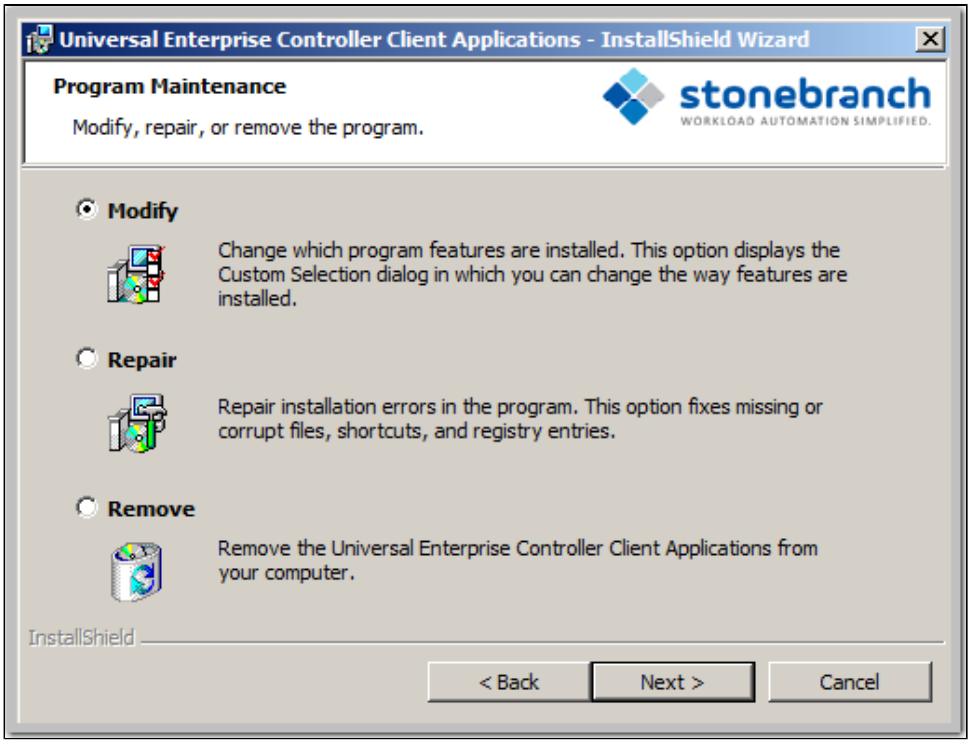
Windows Server 2003

If you are using Windows Server 2003, select **Add or Remove Programs** on the Windows Control Panel.

Step 2	From the list of installed programs, select **Universal Enterprise Controller Client Applications** and click the **Change** button to start Windows Installer. The Welcome dialog displays. 

Step 3

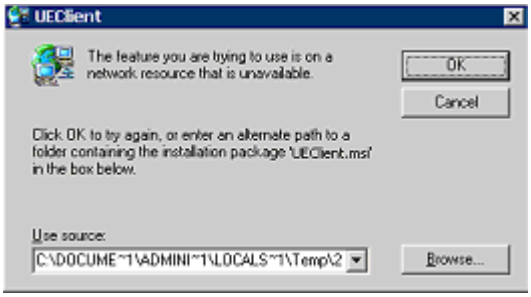
Click the **Next>** button. The Program Maintenance dialog displays.



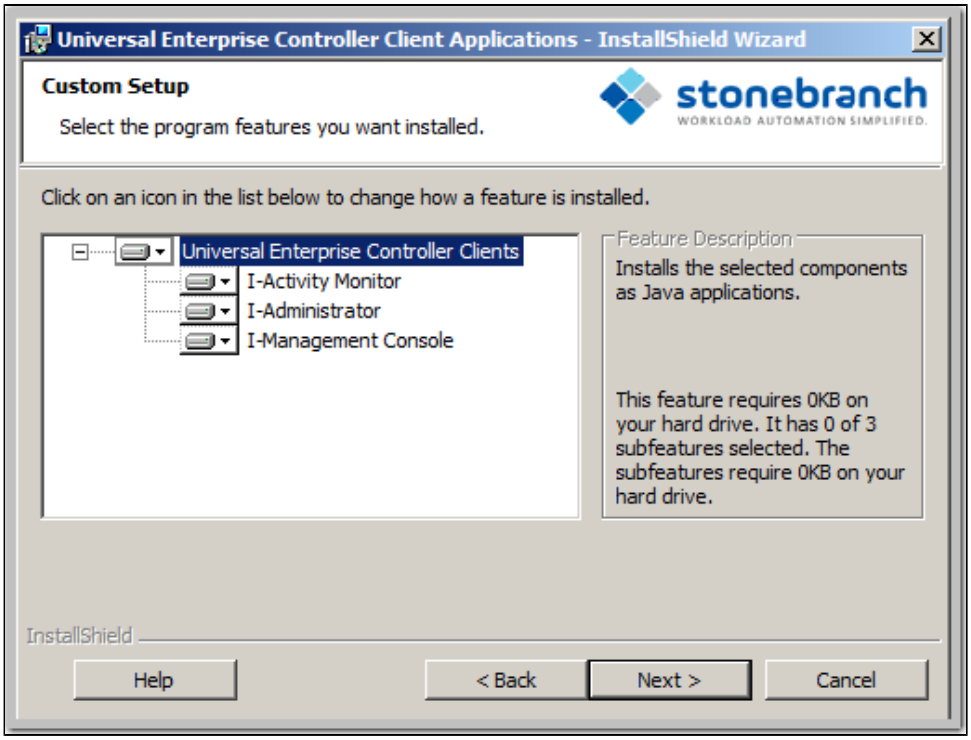
Windows Server 2003

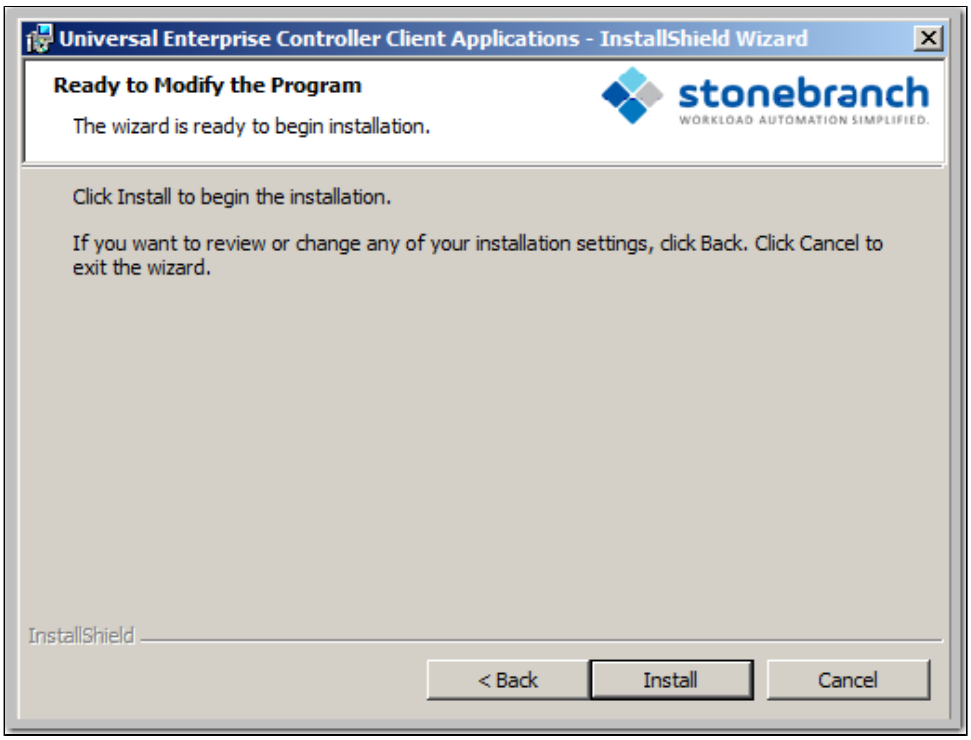
If the UEC Client Applications were installed via a Remote Desktop connection, the environment settings used during that session may no longer be available. Windows Server 2003 allows multiple Remote Desktop sessions for multiple users, and each session has its own environment. Depending on the way the Remote Desktop session for the UEC Client Applications installation was established, some problems may arise whenever an attempt is made to modify that installation.

The distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. Consequently, any attempts to modify the installation may fail because the Windows Installer cannot locate the installation's source files (a dialog similar to the one shown below may be displayed).



To resolve this issue, re-extract the distribution files to a location that is independent of a Remote Desktop environment and specify that location in the dialog above. Keep in mind, however, that the extracted files must come from the same distribution package used to do the initial install. If matching distribution files can't be found, the UEC Client Applications must be uninstalled and then reinstalled with the desired modifications.

Step 5	Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog.  Currently installed components are identified by a drive icon. Uninstalled components are identified by an **X** icon.
Step 6	To remove a currently installed component: 1. Click the drive icon next to that component. 2. Select the X icon from the drop-down list to mark the component for removal.
Step 7	To add an uninstalled component: 1. Click the X icon next to that component. 2. Select the drive icon from the drop-down list to mark the component for installation.

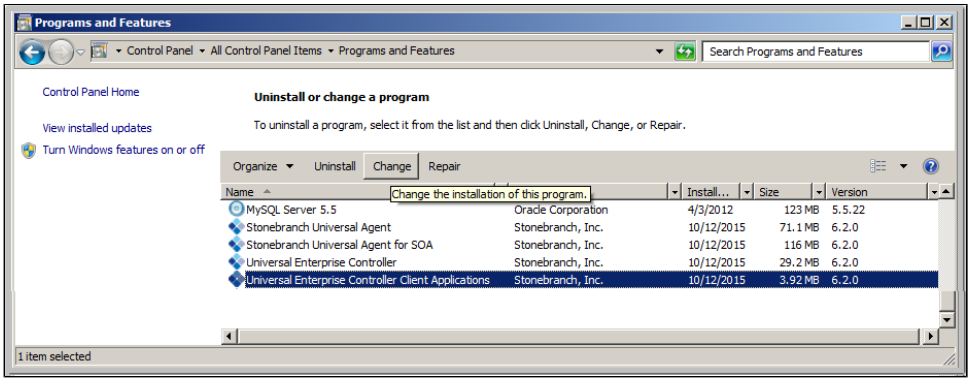
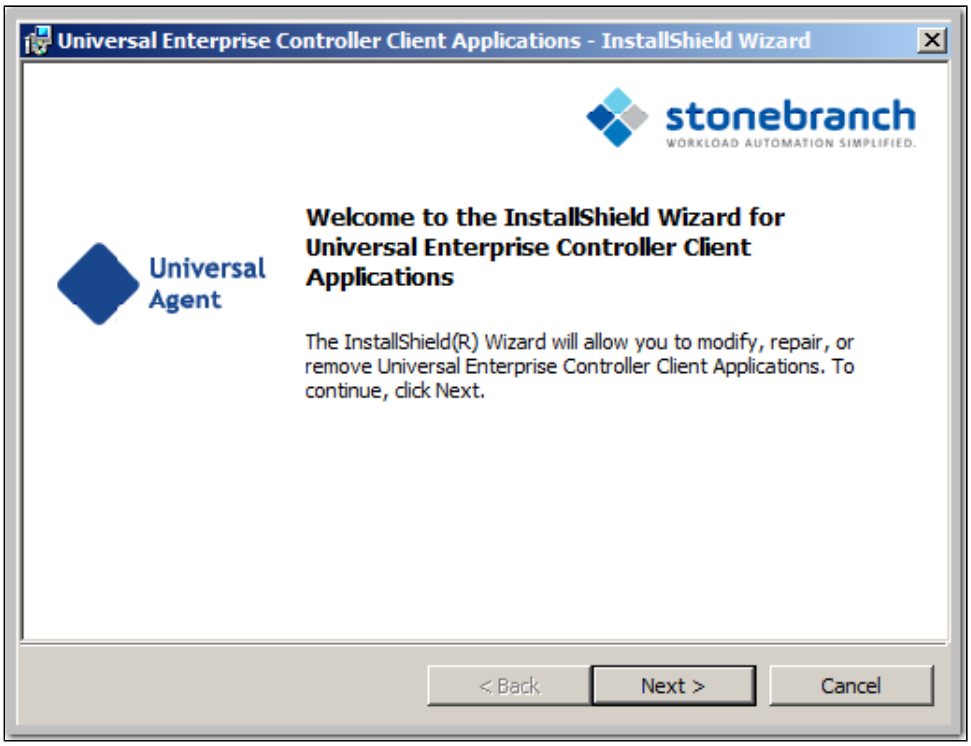
Step 8	Click the **Next>** button to display the Ready to Modify the Program dialog. 

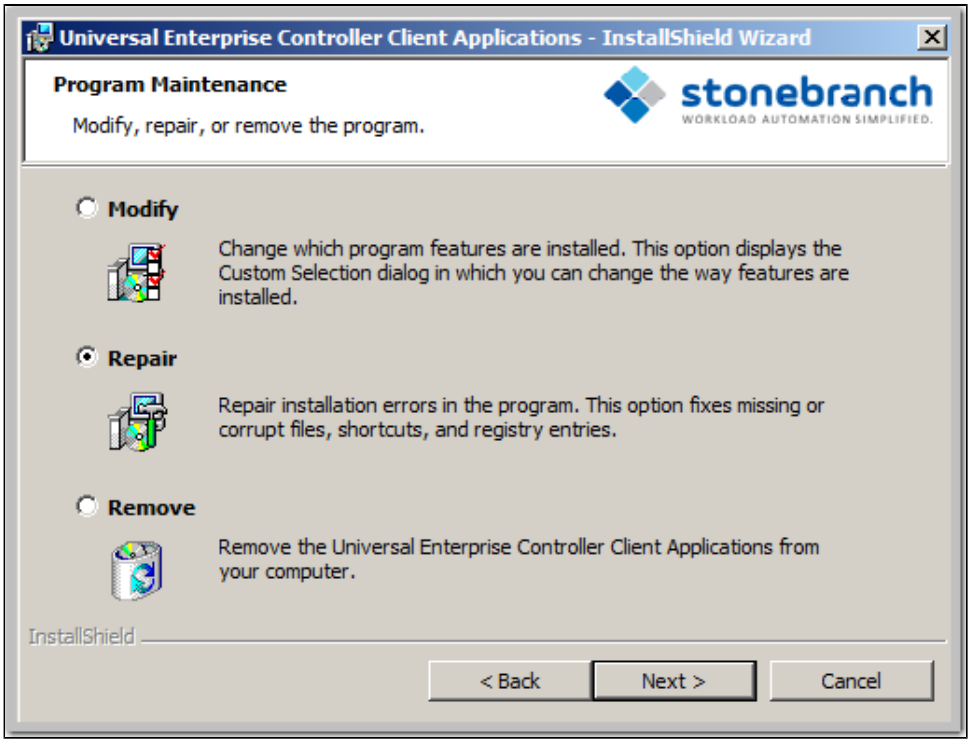
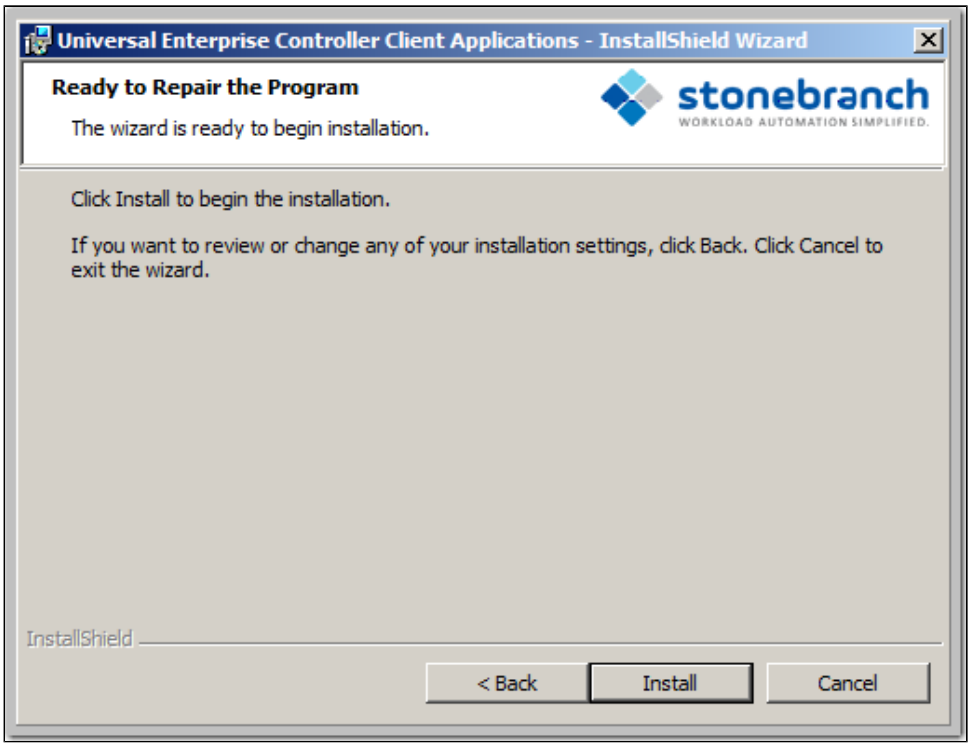
Step 9	Click the **Install** button to modify the installation.
	
	When the modifications are complete, the following actions will be taken:
	
	• Components marked with a drive icon will:
	  • Remain installed if they already are installed.
	  • Be installed if they are not already installed.
	• Components marked with an **X** will:
	  • Remain uninstalled if they are not currently installed
	  • Be removed if they currently are installed.

Repairing a Corrupted UEC Client Applications Installation

Windows Installer has the ability to recover accidentally deleted application files or registry entries required by the UEC Client Applications. This repair feature will re-install the missing items, making a complete re-install unnecessary.

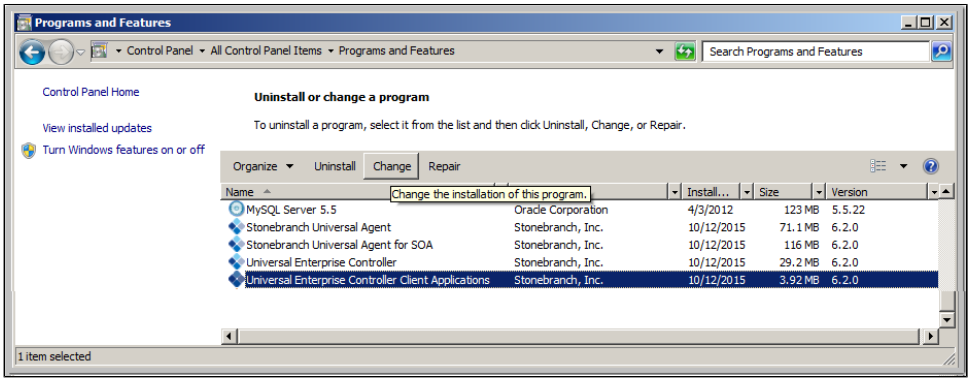
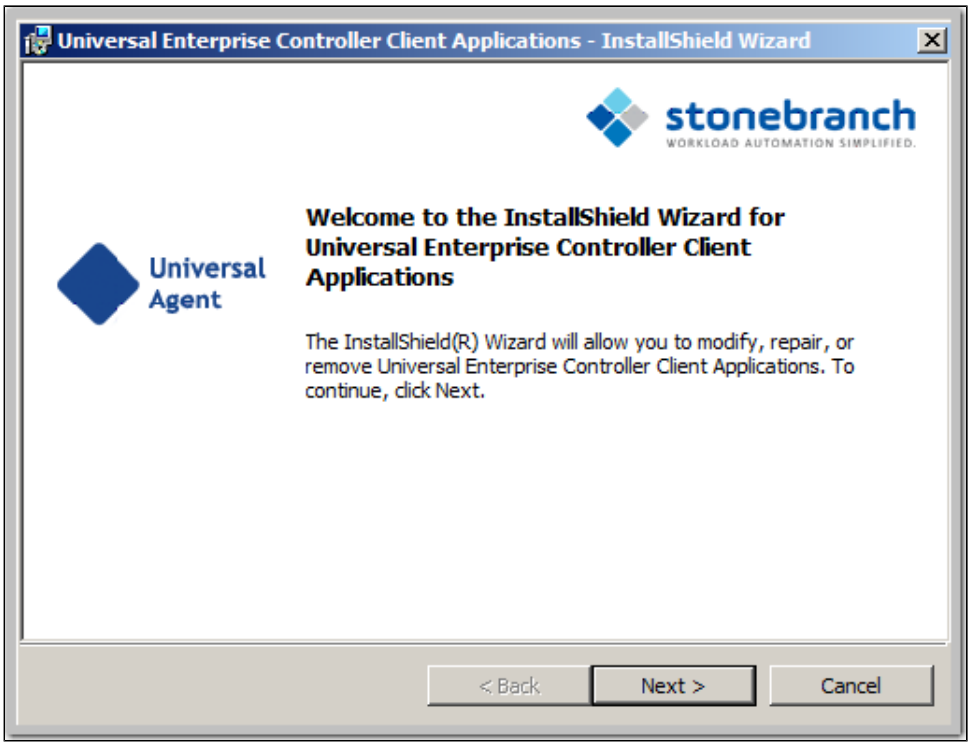
To repair an installation, perform the following steps:

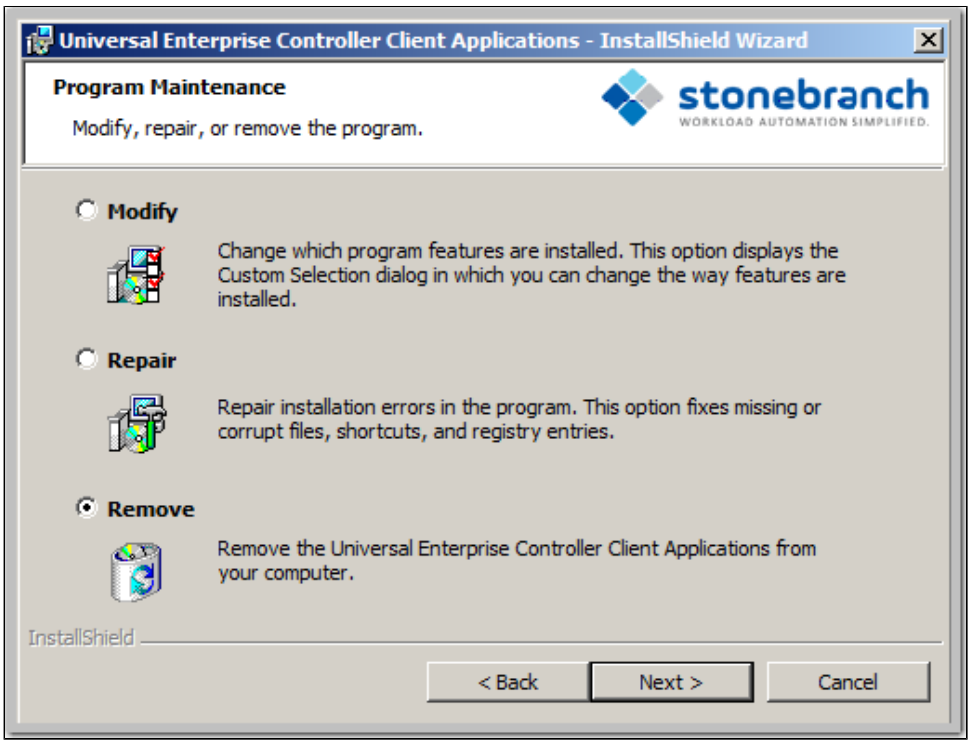
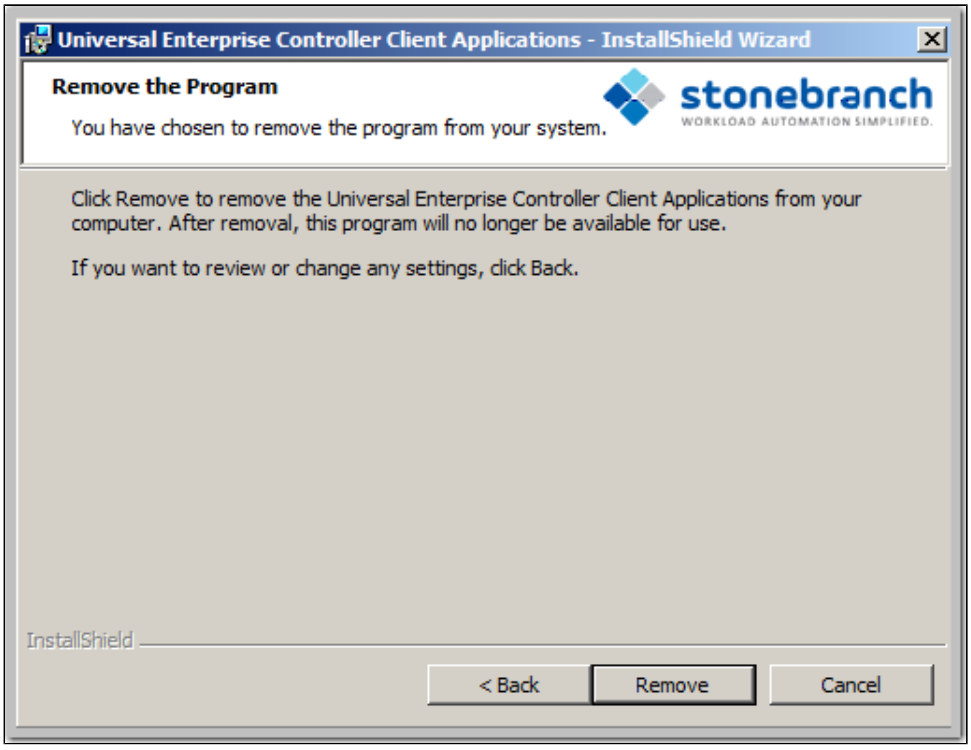
Step 1	On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.  pre-Vista versions of Windows If you are using an earlier version of Windows than Windows Vista, select **Add or Remove Programs** on the Windows Control Panel.
Step 2	From the list of installed programs, select **Universal Enterprise Controller Client Applications** and click the **Change** button to start Windows Installer. The Welcome dialog displays. 

Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 
Step 4	Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair the Program dialog. 
Step 5	Click the **Install** button to repair the installation.

Removing a UEC Client Applications Installation

To uninstall a UEC Client Applications installation, perform the following steps:

Step 1	On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.  pre-Vista versions of Windows If you are using an earlier version of Windows than Windows Vista, select **Add or Remove Programs** on the Windows Control Panel.
Step 2	From the list of installed programs, select **Universal Enterprise Controller Client Applications** and click the **Change** button to start Windows Installer. The Welcome dialog displays. 

Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 
Step 4	Click the **Remove** radio button, and then the **Next>** button, to display the Remove the Program dialog. 
Step 5	Click the **Remove** button to remove the installation.

Un-Installed Files

The uninstall process will remove only those files created during the installation. Some files stored under the **.\Universal** install directory may be left behind after the uninstall. In this situation, those files and/or directories may simply be deleted.

Before deleting the entire **.\Universal** directory, make sure that no other Stonebranch, Inc. products are installed there. (See UEC Client Applications - File Inventory Lists for a list of files and directories installed with UEC Client Applications.)

Installing UEC Client Applications via the Command Line

- Introduction
- Installing UEC Client Applications
  - Command Line Syntax
  - Command Line Switches
  - Command Line Parameters
  - Command Line Installation Examples
- Detecting the Completion of Silent Installs

Introduction

This page describes how to install UEC Client Applications using the Windows Installer command line interface.

A command line installation is useful in situations where:

- Several UEC Client Applications installations must be deployed across many different systems.
- It is not practical or convenient to perform the graphical interface installation.
- It is necessary to generate an installation log file.

Installing UEC Client Applications

Step 1	Download the UEC Client Applications for Windows product distribution file, **`sb-UEClient-6.3.0.<level>-windows-i386.exe`**, to your work station.
Step 2	Execute the distribution file from the command line, and include all appropriate command line switches and parameters. The installation process determines whether a Windows Installer update is needed. The process then extracts and saves a Windows installer package file (.msi) to this location. After all files (including the .msi) are extracted from the distribution file, the installation process verifies that your machine meets the minimum installation requirements. If the requirements are met, the installation begins.

Command Line Syntax

The following illustrates the command line syntax used to install UEC Client Applications:

```
sb-UEClient-6.3.0.<level>-windows-i386.exe [/v"command line parameters"] [/s] [/w] [/x]
```

In this syntax:

- **`<level>`** is the numeric package level.

The command line switches (/v, /s, /w, and /x) are processed directly by the distribution file to control behavior of the Windows Setup application.

The command line parameters are passed to the Windows Installer (**msiexec**) to control the extracted Windows installer package file (.msi) behavior during the install process.

Command Line Switches

The following table identifies the command line switches available for a command line installation:

/v	Passes parameters to the Windows Installer (**msiexec**). The list of parameters must be enclosed in double (") quotation marks. See Command Line Parameters for available parameters.
/s	Suppresses the initialization and extraction dialogs displayed before the product install Welcome dialog." If you are using the /q command line parameter, use this switch additionally for a completely silent install.

/w	Instructs the Windows Setup application to wait until the installation completes. Use this switch when launching the installation from a script file. Without it, the Setup application may return immediately after launching Windows Installer.
/x	Uninstalls UEC Client Applications.

Command Line Parameters

The following table describes the parameters that are available for a command line installation.

The parameters can be specified in any order, with the following exceptions:

- If the Repair (/fom) or Remove (/x) parameter is used, it must be specified ***before*** all other parameters.
- If the Silent install (/q) and/or Log file (/L) parameters are used, they can be specified in any order, but they must be specified ***after*** all other parameters.

These parameters are preceded by the /v command line switch and must be enclosed in double (") quotation marks.

Parameter	Description	Default
/fom	Repairs a UEC Client Applications installation. **om** (after the **/f**) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the **om** options must be used. **/fom** cannot be used with the **/x** (remove) parameter.	n/a
/x	Removes the UEC Client Applications. **/x** cannot be used with the **/fom** (repair) parameter.	n/a
INSTALLTO=*installdir*	Sets the root installation directory to <**`installdir`**>. Each component will be installed under this directory. **INSTALLTO** is required if you want to install the UEC Client Applications in a directory that is different from the default, which varies depending on the type of installation being done (see PERUSER parameter). If the directory contains spaces, you must use double (**"**) quotation marks around the path name.	**per-machine installation**: Directory specified by the **PROGRAMFILES** environment variable (typically **`C:\Program Files\Universal`**). **per-user installation**: Location under the directory specified by the **USERPROFILE** environment variable. For example, if the **USERPROFILE** directory is set to **`C:\Documents and Settings\username`**, the default target directory will be **`C:\Documents and Settings\username\Application Data\Universal`**. If the UEC Client Applications already are installed, the installation directory will default to its existing location.
PERUSER={**0**|**1**}	Performs an installation for all users or a specific user account: • **0** specifies a per-machine installation; it performs an install for all users of a given machine. This means that any UEC Client Applications configuration options stored in the Windows registry, Start menu short cuts that launch the UEC Client Applications, and the applications files themselves will be visible to all accounts on the machine where the UEC Client Applications was installed. • **1** specifies a per-user installation; it performs an install that is specific to the user account doing the installation. This means that any UEC Client Applications configuration options stored in the Windows registry, Start menu short cuts that access the UEC Client Applications, and the application files themselves will be visible only to the user account with which the installation was performed. It will appear to all other users of a given machine as though the UEC Client Applications is not actually installed. **PERUSER** is required only under the following situations: • For per-user installs, when the installation is being done with a Windows account that is a member of the Administrators group. • For uninstalls, where a Windows account that is a member of the Administrators group is removing a per-user installation.	**0**, if UEC Client Applications is installed using an Administrative account. **1**, if a regular user account (that is, a non-Administrative user) is executing the installation.

UECADMIN ={**yes**|**no**}	Specification for whether or not to install the I-Administrator component during new installs, upgrades, or maintenance. • If **yes** is specified, I-Administrator will be installed. • If **no** is specified, I-Administrator will not be installed. If I-Administrator already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UECADMIN** is not required unless you want to change the current install state. **UECADMIN** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **I-Administrator** on the Custom Setup dialog when installing UEC Client Applications via the graphical interface.	**yes**
UAMONITOR ={**yes**|**no**}	Specification for whether or not to install the I-Activity Monitor component during new installs, upgrades, or maintenance. • If **yes** is specified, I-Activity Monitor will be installed. • If **no** is specified, I-Activity Monitor will not be installed. If I-Activity Monitor already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UAMONITOR** is not required unless you want to change the current install state. **UAMONITOR** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **I-Activity Monitor** on the Custom Setup dialog when installing UEC Client Applications via the graphical interface.	**yes**
UMGMTCON ={**yes**|**no**}	Specification for whether or not to install the I-Management Console component during new installs, upgrades, or maintenance. • If **yes** is specified, I-Management Console will be installed. • If **no** is specified, I-Management Console will not be installed. If I-Management Console already is present on the system (via a previous installation), it will be removed. Since, by default, each component's install state is preserved during an upgrade or maintenance, **UMGMTCON** is not required unless you want to change the current install state. **UMGMTCON** is ignored during an uninstall. Setting this parameter has the same effect as selecting whether or not to install **I-Management Console** on the Custom Setup dialog when installing UEC Client Applications via the graphical interface.	**yes**
/q	Suppresses the product installation dialogs. Use this parameter in addition to the /s command line switch for a completely silent install. See Command Line Switches, Command Line Installation Examples, and Detecting the Completion of Silent Installs for additional information regarding silent installs.	n/a
/L*v	Instructs the installation process to create an installation log file named <`logfilepath`> (full path name). If <`logfilepath`> contains spaces, you must enclose it with double (`"`) quotation marks around the path name. ***v** are flags used to specify the level of detail (**verbose**) contained in the log file. To reduce the amount of output generated, ***v** can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.	n/a

Command Line Installation Examples

The following examples illustrate different ways that UEC Client Applications can be installed from the command line.

Graphical User Interface Install of All Components

To install all UEC Client Applications components via the graphical user interface, issue the following command:

```
sb-UEClient-6.3.0.x-windows-i386.exe
```

Graphical User Interface Install, All Components, with Log File

To install all UEC Client Applications components using the Windows Installer graphical user interface and write a log file to **`C:\temp\install.log`** during the installation, issue the following command:

```
sb-UEClient-6.3.0.x-windows-i386.exe /v"/l*v c:\temp\install.log"
```

Silent Install of All Components

To silently install all UEC Client Applications components, issue the following command:

```
sb-UEClient-6.3.0.x-windows-i386.exe /s /v"/qn"
```

Silent Install of All Components, Per-User Installation, Different Installation Directory

To silently install all UEC Client Applications components as a *per-user* installation (that is, one in which registry entries, Start menu shortcuts, and so on are visible only to the user performing the installation), and to override the default installation directory, issue the following command:

```
sb-UEClient-6.3.0.x-windows-i386.exe /s /v"/qn PERUSER=1 INSTALLTO=C:\UEClientApps\Universal"
```

Silent Install of All Components, with Log File

To silently install all UEC Client Applications components and write a log file to **C:\Temp\install.log** during the installation, issue the following command:

```
sb-UEClient-6.3.0.x-windows-i386.exe /s /v"/qn /l*v c:\temp\install.log"
```

Detecting the Completion of Silent Installs

If the **/q** switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that can be used to detect when the Installer process (**msiexec.exe**) ends is to execute it using the system's **start** command. Using available command line switches, the **start** command can be used to kick off the Installer process, and then wait for it to finish. When the **start** command returns control to its calling process (for example, the command prompt), the Installer process will have ended.

For example, from the command prompt, the following command can be issued to start the UEC Client Applications installation, and wait for it to finish.

```
start /b /wait sb-UEClient-6.3.0.x-windows-i386.exe /w /s /v"/qn"
```

- The **/b** switch prevents the **start** command from opening a new window.
- The **/wait** parameter causes the **start** command to start the application, **sb-UEClient-6.3.0.x-windows-i386.exe**, and then wait for it to finish.

The syntax above also can be used to execute the **start** command from within a script, such as a .**bat** file.

For more information on the **start** command, go to the Windows command prompt and enter: **start /?**.

Modifying a UEC Client Applications Installation via the Command Line

- Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface
- Adding or Removing UEC Client Applications Components
- Repairing a Corrupted UEC Client Applications Installation
- Removing UEC Client Applications from the Command Line

Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface

This page describes how to modify a UEC Client Applications installation via the Windows Installer command line interface.

After UEC Clients are is installed, Windows Installer can be run as many times as needed to modify the installation by:

- Adding or Removing UEC Client Applications Components
- Repairing a Corrupted UEC Client Applications Installation
- Removing a UEC Client Applications Installation

(For a description of the parameters used in these procedures, see Windows Installer Command Line Parameters.)

Adding or Removing UEC Client Applications Components

Currently, it only is possible to add or remove UEC Client Applications components using the Windows Installer graphical interface. (see **Adding or Removing Components** in Modifying a UEC Client Applications Installation via the Graphical Interface).

Repairing a Corrupted UEC Client Applications Installation

To recover accidentally deleted files or registry entries required by the UEC Client Applications using the Windows Installer command line interface, use the **/f** switch together with the **om** parameters.

These are the same repair options set internally by the graphical interface installation. They cause Windows Installer to reinstall files that are missing or that are older than the version contained in the UEC Client Applications distribution file.

Silent Repair

To repair a UEC Client Applications installation from the command line, without using the Windows Installer graphical interface, issue the following command:

```
msiexec.exe /fom SetupPath\UEClient.msi /q
```

Interactive Repair, with Log File

To repair a UEC Client Applications installation using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to **C:\Temp\repair.log** while running the repair, issue the following command:

```
msiexec.exe /fom SetupPath\UEClient.msi /L C:\Temp\repair.log
```

Removing UEC Client Applications from the Command Line

To uninstall UEC Client Applications using the Windows Installer command line interface, use the **/x** command line switch.

Silent Removal

To uninstall the UEC Client Applications without using a graphical interface, issue the following command:

```
msiexec.exe /x SetupPath\UEClient.msi /q
```

Silent Removal, Per-User Installation

To remove a *per-user* installation that was installed using an Administrator account, issue the following command:

```
msiexec.exe /x SetupPath\UEClient.msi PERUSER=1 /q
```

A Stonebranch Tip

If you know that the UEC Client Applications were installed using an account that is a member of the Administrators group, but are not sure if a *per-user* or *per-machine* installation was done, it might be best to uninstall the Client Applications using the graphical interface (see Removing a UEC Client Applications Installation).

This will ensure complete removal of the UEC Client Applications.

If you are removing a UEC Client Applications installation that was installed using a non-administrative account, the **PERUSER** parameter is not necessary.

UEC Client Applications - 64-Bit Windows Editions

UEC Client Applications - Installing on 64-bit Windows Editions

All Universal Agent components have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

The installation defaults for the UEC Client Applications should not require any modification when installing on 64-bit Windows editions.

UEC Client Applications - File Inventory Lists

- UEC Client Applications - File Inventory List
  - System 32 Path
- I-Administrator
- I-Activity Monitor
- I-Management Console
- System Files

UEC Client Applications - File Inventory List

The UEC Client Applications installation includes the files required for the following components:

- I-Administrator
- I-Activity Monitor
- I-Management Console

If any of the components already are installed, Windows Installer will upgrade them to the latest version.

This page lists the files installed with each UEC Client Applications component. The file paths specified are relative to the root installation directory that was specified during the installation.

System 32 Path

Items shown with a path of System32 are installed in the 32-bit system folder. The actual name of this directory depends on the Windows version:

- For all supported 32-bit Windows editions, the path is **\Windows\System32**.
- For all supported 64-bit Windows editions, the path is **\Windows\SysWow64**.

I-Administrator

File	Description
UECAdmin\uecadmin.jar	I-Administrator Java archive (JAR) file.
UECAdmin\lib\ueccommon.jar	Common routines shared between I-Administrator and I-Management Console.
UECAdmin\lib\uecumc.jar	I-Management Console resources used by I-Administrator.

I-Activity Monitor

File	Description
UAMonitor\uam.jar	I-Activity Monitor Java archive (JAR) file.

I-Management Console

File	Description
UMConsole\uecumc.jar	I-Management Console Java archive (JAR) file.
UMConsole\lib\ueccommon.jar	Common routines shared between I-Management Console and I-Administrator.

System Files

The following files will be installed only if they are newer than the existing file.

File	Description
System32\msiexec.exe	Version 3.1.4000.1823 of the Microsoft Windows Installer (see Windows Installer for more information).

Universal Agent for SOA for Windows Installation

Universal Agent for SOA on Windows is now available as 6.9.0.0. It is a 64-bit package, supported on the same Windows versions as the Agent: Server 2012, Server 2012 R2, Server 2016, and Server 2019.

These pages provide information for that version.

Note

Starting with the 6.7.0.0 release, the Universal Agent for SOA on Windows is a 64-bit application that replaces all previous versions. There is no upgrade path from the older 32-bit version to the 64-bit implementation. If the install detects an existing 32-bit version, it will prompt you to remove that version before proceeding.

You are responsible for preserving your existing configuration in this scenario. PLEASE remember to collect the license information stored in the `uacs.conf` file so that you can re-apply it following the 64-bit install. If you lose your license information, you will need to contact Stonebranch customer support to retrieve that information.

See also Universal Agent for SOA for Windows - Pre-Installation - Upgrade Backups.

Overview

The following information is provided for the installation of Universal Agent for SOA for Windows:

- Installation Package
- Installation Requirements
- Pre-Installation - Upgrade Backups
- Installation Procedures
- Configuring and Starting UAC Server
- 64-Bit Windows Editions
- File Inventory Lists

(For licensing information, see Windows Installation - Licensing.)

Universal Agent for SOA for Windows - Installation Package

Universal Agent for SOA on Windows is now available as 6.7.0.0. It is a 64-bit package, supported on the same Windows versions as the Agent: Windows 7, Server 2008 R2, Server 2012, Server 2012 R2, Server 2016, and Server 2019.

Note

Starting with the 6.7.0.0 release, the Universal Agent for SOA on Windows is a 64-bit application that replaces all previous versions. There is no upgrade path from the older 32-bit version to the 64-bit implementation. If the install detects an existing 32-bit version, it will prompt you to remove that version before proceeding.

You are responsible for preserving your existing configuration in this scenario. PLEASE remember to collect the license information stored in the `uacs.conf` file so that you can re-apply it following the 64-bit install. If you lose your license information, you will need to contact Stonebranch customer support to retrieve that information.

See also Universal Agent for SOA for Windows - Pre-Installation - Upgrade Backups.

Components

The Universal Agent for SOA 6.7.0 for Windows package includes the following components:

- Universal Agent for SOA
  - Universal Application Container Server
  - Universal Application Container
  - Universal Application Interface

Component Compatibility

The following table identifies the compatibility of Universal Agent for SOA 6.7.0 for Windows with previous component / product versions.

Component	Compatibility
Universal Agent for SOA 6.7.0	Universal Command Manager 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, and 3.2.0.

The component references pertain to all supported platforms for that version.

Note

The 64-bit version of the SOA package (6.7.0) will work only with 64-bit versions of Universal Command Server.

Universal Agent for SOA for Windows - Installation Requirements

Universal Agent for SOA on Windows is now available as 6.7.0.0. It is a 64-bit package, supported on the same Windows versions as the Agent: Windows 7, Server 2008 R2, Server 2012, Server 2012 R2, Server 2016, and Server 2019.

Note

Starting with the 6.7.0.0 release, the Universal Agent for SOA on Windows is a 64-bit application that replaces all previous versions. There is no upgrade path from the older 32-bit version to the 64-bit implementation. If the install detects an existing 32-bit version, it will prompt you to remove that version before proceeding.

You are responsible for preserving your existing configuration in this scenario. PLEASE remember to collect the license information stored in the `uacs.conf` file so that you can re-apply it following the 64-bit install. If you lose your license information, you will need to contact Stonebranch customer support to retrieve that information.

See also Universal Agent for SOA for Windows - Pre-Installation - Upgrade Backups.

Additional Requirements

In addition, you must have:

- An account with administrative privileges.
- Possible reboot. A reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1283 is installed, or required files are in use at the time of the installation.
- TCP/IP.
- About 110 megabytes of disk space.
- A supported 64-bit Universal Agent.

Note

Starting with the 6.7.0.0 release, the Universal Agent for SOA on Windows is a 64-bit application that replaces all previous versions. There is no upgrade path from the older 32-bit version to the 64-bit implementation. If the install detects an existing 32-bit version, it will prompt you to remove that version before proceeding.

You are responsible for preserving your existing configuration in this scenario. PLEASE remember to collect the license information stored in the `uacs.conf` file so that you can re-apply it following the 64-bit install. If you lose your license information, you will need to contact Stonebranch customer support to retrieve that information.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Universal Agent for SOA for Windows - Pre-Installation - Upgrade Backups

Universal Agent for SOA for Windows - Pre-Installation / Upgrade Backups

Before upgrading to the latest release of Universal Agent for SOA, we recommend stopping any active Universal Application Container (UAC) Server instances (via the Universal Control utility or by stopping the local Universal Broker). If the UAC Server is active during the upgrade, the Windows Installer will require a reboot of the system. Also, stopping the UAC Server before backing up the files listed below will ensure the latest copies of each are captured.

The installation process overwrites the current files (exception: see the Note for Log4jConfiguration.xml in the following table), effectively removing your modifications. Backing up these files will optimize the time it takes you to get up and running after installing or upgrading.

The following table identifies the files - and their locations - that should be backed up or copied before you install a new release or upgrade a current release.

File	Location
UAC.xml File	**`%Program Data%\Universal\uac`**
Log4jConfiguration.xml File	**`Program Files\Universal\uac`** (UAC) **`Program Files\Universal\uai`** (UAI) Note The installation process does not overwrite Log4jConfiguration.xml files.
JMS Provider Client Jar Files	**`Program Files\Universal\uac\container\webapps\axis2 \WEB-INF\lib`** Note The names of the jar files differ depending on which JMS Provider you are using.
JMS Provider Client Properties Files	**`Program Files\Universal\uai\xml`** Note These are suggested locations; you can place these files anywhere on the files system. If you have located these files under the **uai** directory, you should back them up.
Payload Files	Normally, payload files should be located on the host system where Universal Command Manager is installed. If you have chosen to store them elsewhere, the suggested location is: **`Program Files\Universal\uai\xml`** Note You can place these files anywhere on the files system. If you have located these files under the **uai** directory, you should back them up.

Universal Agent for SOA for Windows - Installation Procedures

Universal Agent for SOA for Windows - Installation Procedures

The following procedures are provided for the installation and modification of Universal Agent for SOA for Windows:

- Installing SOA for Windows via the Graphical Interface
- Modifying a SOA for Windows Installation via the Graphical Interface
- Installing SOA for Windows via the Command Line
- Modifying a SOA for Windows Installation via the Command Line

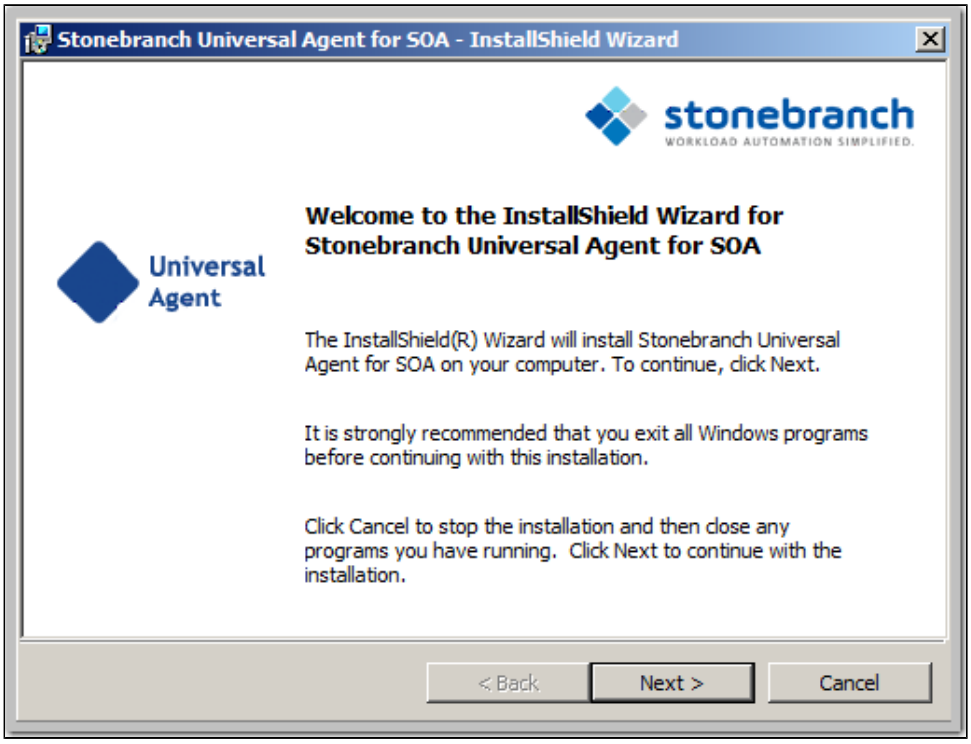
Note

Modifying a Universal Agent for SOA installation refers to the adding / removing of Universal Agent for SOA components, repairing a corrupted installation, or removing an installation. To change the installed version of Universal Agent for SOA, see Upgrading Universal Agent and Applying Maintenance to Universal Agent.

Installing SOA for Windows via the Graphical Interface

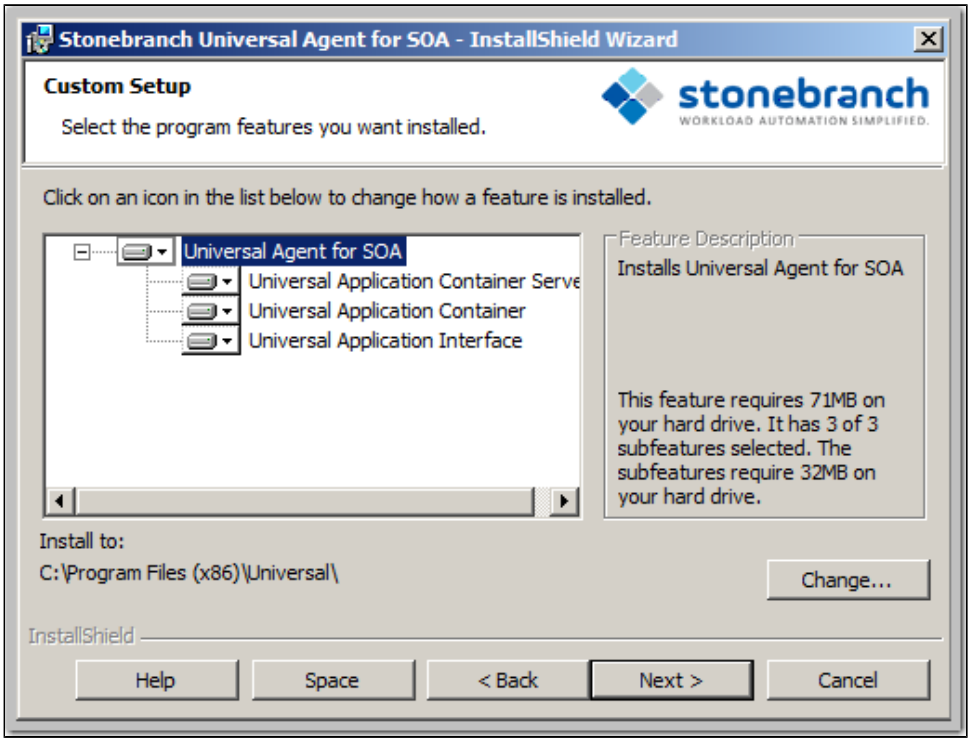
Installing Universal Agent for SOA via the Windows Installer Graphical Interface

To install Universal Agent for SOA for Windows using the Windows Installer graphical interface, perform the following steps:

Step 1	Download the Universal Agent for SOA for Windows product distribution file, **sb-soa-6.7.0.<level>-windows-x64.exe**, to your work station.
Step 2	Execute the distribution file to extract the files. Note If you already have extracted the files from the distribution file, but cancelled the installation process in order to install Windows Installer separately (see Windows Installer), you can simply double-click the extracted Universal Agent for SOA installation file, **Stonebranch Universal Agent for SOA (x64).msi**, to begin the installation. The installation automatically begins after the file extraction completes. It first will verify that your machine meets the minimum system requirements (see Installation Requirements). If the requirements are met, a Welcome dialog displays. 

Step 3

Click the **Next** button. A list of Universal Agent for SOA components included in the installation package then displays. It is from this list that you can select which components to install.



For a new installation, a drive icon displays next to each item in the list, indicating that the component will be installed.

Note

Starting with the 6.7.0.0 release, the Universal Agent for SOA on Windows is a 64-bit application that replaces all previous versions. There is no upgrade path from the older 32-bit version to the 64-bit implementation. If the install detects an existing 32-bit version, it will prompt you to remove that version before proceeding.

You are responsible for preserving your existing configuration in this scenario. PLEASE remember to collect the license information stored in the `uacs.conf` file so that you can re-apply it following the 64-bit install. If you lose your license information, you will need to contact Stonebranch customer support to retrieve that information.

Step 4

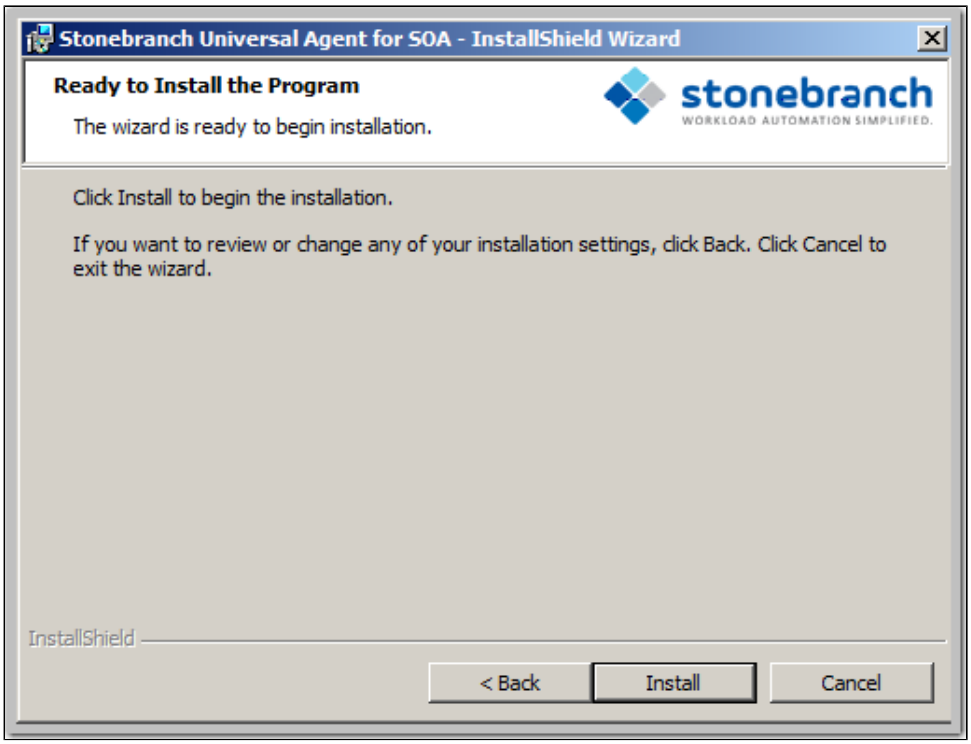
The previous figure shows that all Universal Agent for SOA components will be installed in their respective directories under the **C:\Program Files\Universal** directory.

1. If you want to select a different location, click the **Change...** button.
2. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.

Step 5

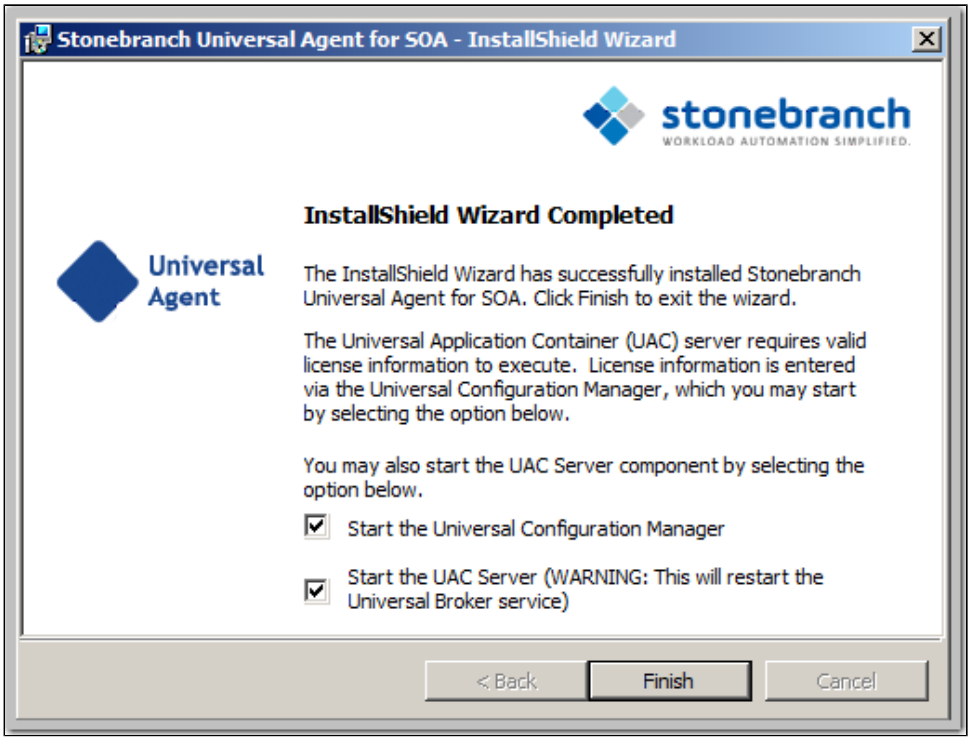
After verifying the install location, click the **Next>** button to continue the installation process.

When the installation is ready to begin, the Ready to Install dialog displays.



Click the **Install** button to begin the installation or click the **<Back** button to return to change information on any of the previous dialogs.

When the installation completes successfully, the Installation Complete dialog displays.



Step 6	If the Universal Application Container (UAC) Server was installed, the following options display on this dialog: • **Start the Universal Configuration Manager** • **Start the UAC Server** You must enter license information into the Universal Application Container Server's configuration before the server will run. Select both of these options to enter the license information and then start the UAC server.
Step 7	Click the **Finish** button to exit the Windows installation.

Modifying a SOA for Windows Installation via the Graphical Interface

- Modifying a Universal Agent for SOA Installation via the Windows Installer Graphical Interface
- Repairing a Corrupted Universal Agent for SOA Installation
- Removing a Universal Agent for SOA Installation
  - Un-Installed Files

Modifying a Universal Agent for SOA Installation via the Windows Installer Graphical Interface

This section describes how to modify a Universal Agent for SOA installation via the Windows Installer graphical interface.

After installing the Universal Agent for SOA, run the installation programs as needed to modify the installation by:

- Repairing a corrupted Universal Agent for SOA Installation.
- Removing a Universal Agent for SOA Installation.

Note

Although Windows Installer provides a Modify selection for Universal Agent for SOA, it cannot be modified; that is, individual components cannot be added or removed.

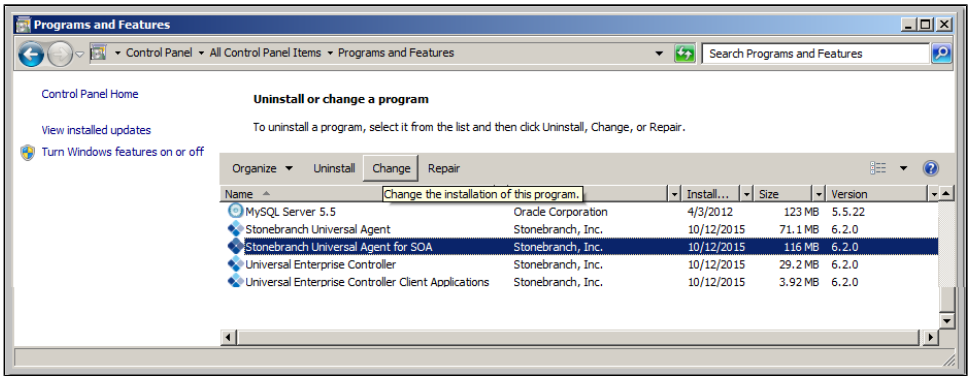
Repairing a Corrupted Universal Agent for SOA Installation

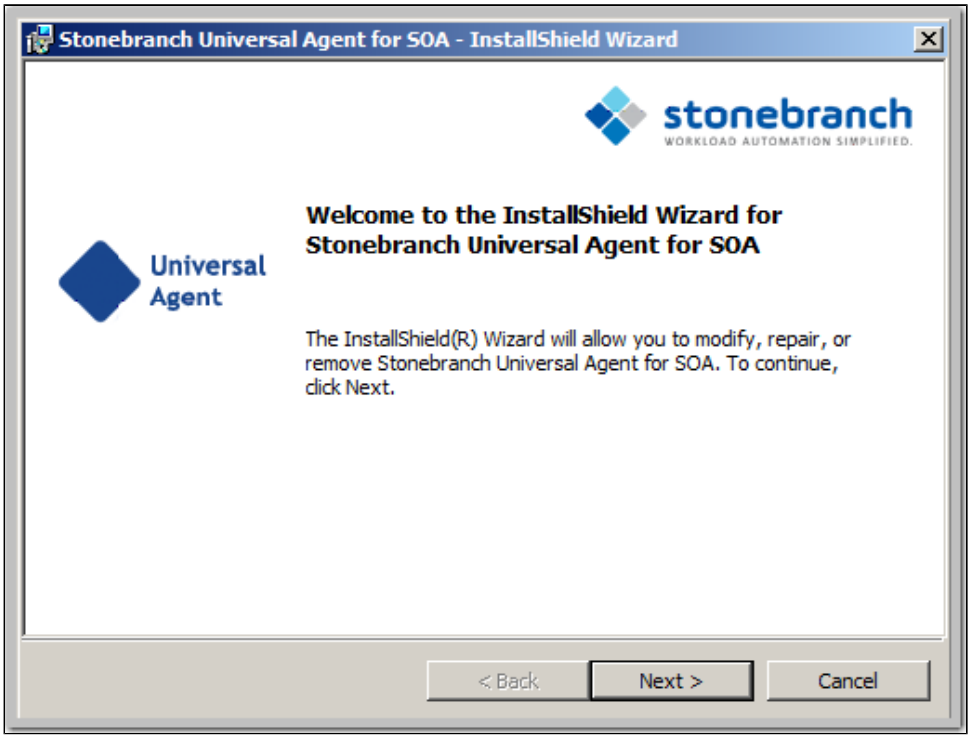
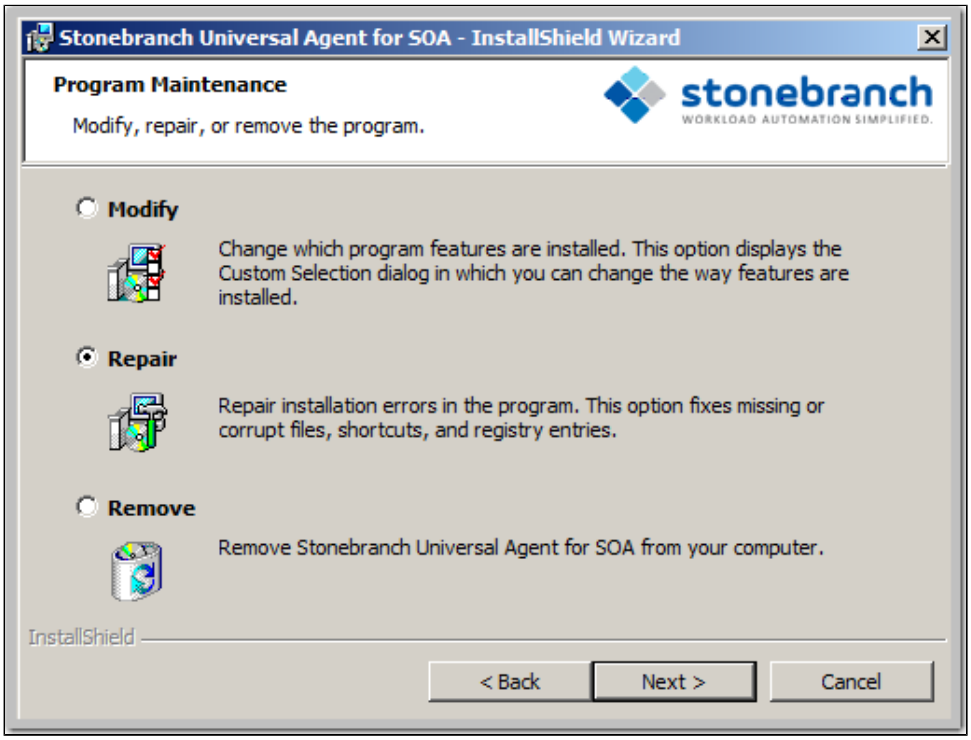
The installation program has the ability to recover accidentally deleted application files, configuration and component definition files, or registry entries required by Universal Agent for SOA. This repair feature will re-install the missing items, making a complete re-install unnecessary.

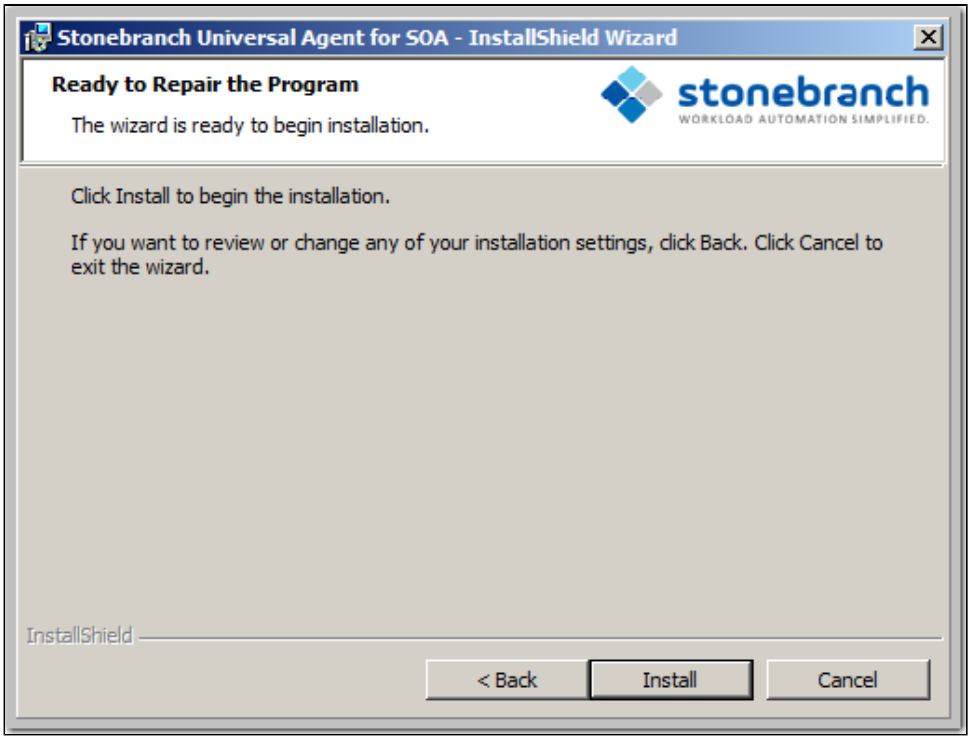
During a repair, any options stored in configuration and component definition files are preserved. If a configuration or component definition file was deleted, the installation will create a new file with default values.

To repair an installation, perform the following steps:

Step 1	On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays.

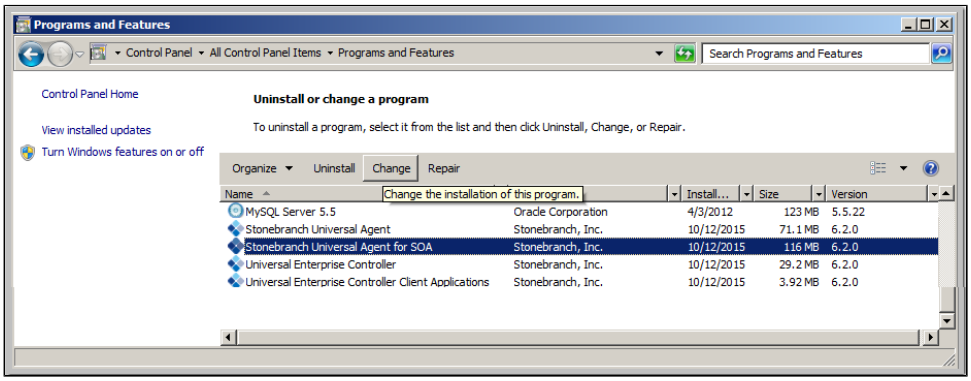


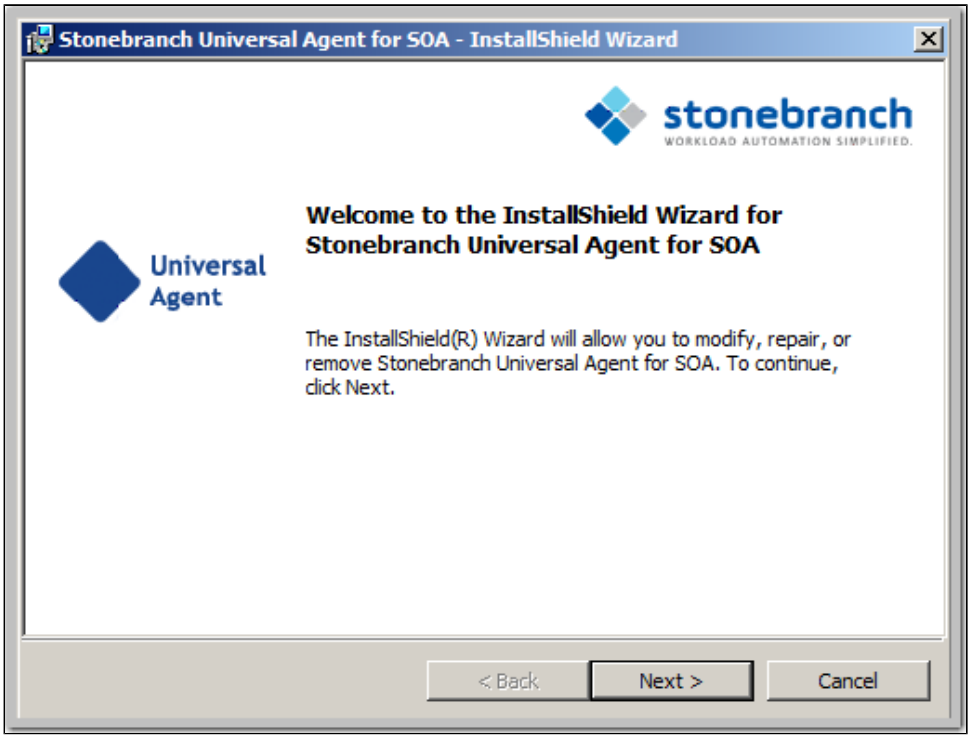
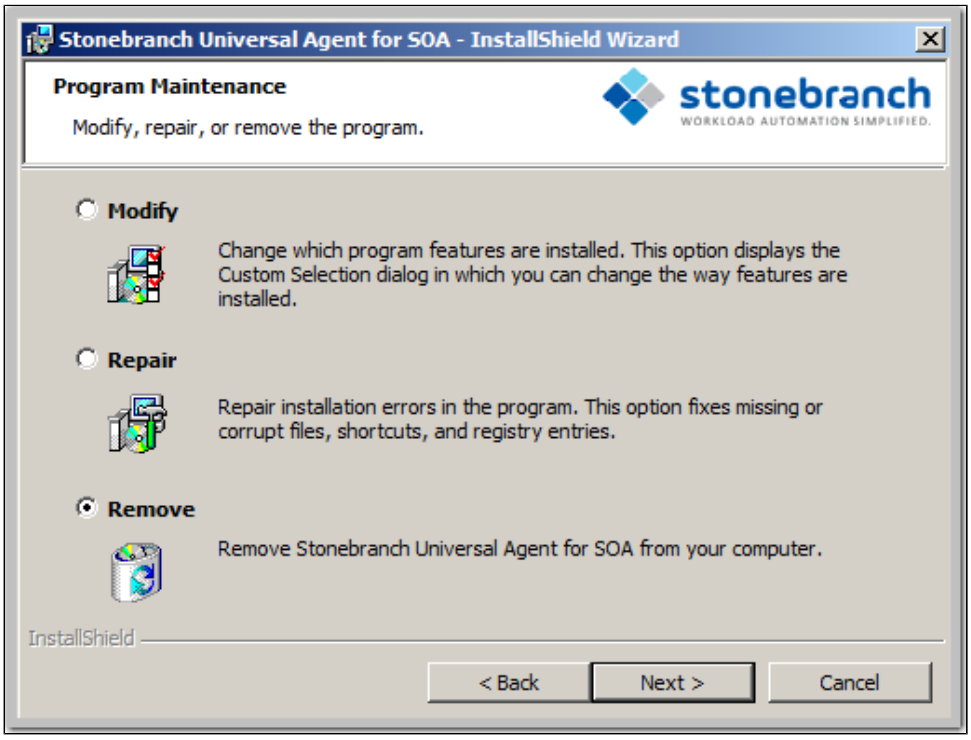
Step 2	From the list of installed programs, select **Stonebranch Universal Agent for SOA** and click the **Change** button to start Windows Installer. The Welcome dialog displays. 
Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 

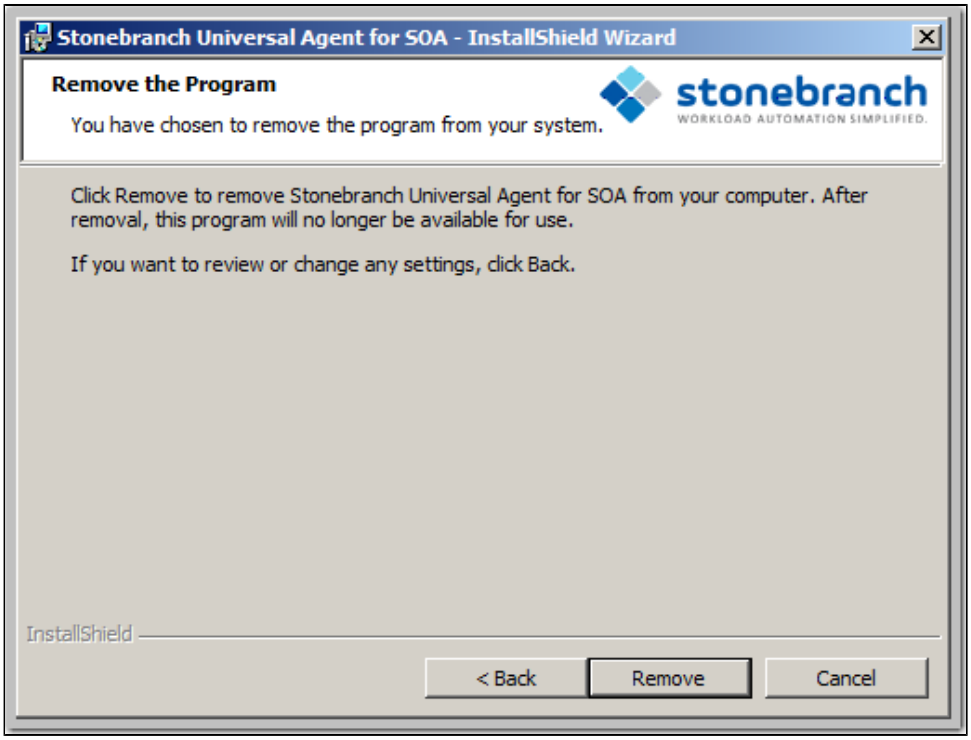
Step 4	Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair the Program dialog. 
Step 5	Click the **Install** button to repair the installation.

Removing a Universal Agent for SOA Installation

To uninstall a Universal Agent for SOA installation, perform the following steps:

Step 1	On the Windows Control Panel, select **Programs and Features**. The Programs and Features dialog displays. 

Step 2	From the list of installed programs, select **Stonebranch Universal Agent for SOA** and click the **Change** button to start Windows Installer. The Welcome dialog displays. 
Step 3	Click the **Next>** button. The Program Maintenance dialog displays. 

Step 4	Click the **Remove** radio button, and then the **Next>** button, to display the Remove the Program dialog. 
Step 5	Click the **Remove** button to remove the installation.

Un-Installed Files

The uninstall process will remove only those files created during the installation. Some files stored under the **.\Universal** install directory by Universal Agent for SOA, such as trace files, may be left behind after the uninstall. In this situation, simply delete the uninstalled files and/or directories.

Before deleting the entire **.\Universal** directory, make sure it does not contain application files for any other installed Stonebranch, Inc. product. (See Universal Agent for SOA for Windows - File Inventory Lists for a list of files and directories installed with Universal Agent for SOA.)

Installing SOA for Windows via the Command Line

- Introduction
- Installing Universal Agent for SOA
  - Command Line Syntax
  - Command Line Switches
  - Command Line Parameters
  - Command Line Installation Examples
- Detecting the Completion of a Silent Install

Introduction

This page describes how to install Universal Agent for SOA using the Windows Installer command line interface.

A command line installation is useful in situations where:

- Several Universal Agent for SOA installations must be deployed across many different systems.
- It is not practical or convenient to perform a graphical interface installation.
- It is necessary to generate an installation log file.

Installing Universal Agent for SOA

Step 1	Download the Universal Agent for SOA for Windows product distribution file, **`sb-soa-6.7.0.<level>-windows-x64.exe`**, to your work station.
Step 2	Execute the distribution file from the command line, and include all appropriate command line switches and parameters. The installation process determines whether a Windows Installer update is needed. The process then extracts and saves a Windows installer package file (.msi) to this location. After all files (including the .msi) are extracted from the distribution file, the installation process verifies that your machine meets the minimum installation requirements. If the requirements are met, the installation begins.

Command Line Syntax

The following illustrates the command line syntax used to install Universal Agent for SOA:

```
sb-soa-6.7.0.-windows-x64.exe [/v"command line parameters"] [/s] [/w] [/x]
```

In this syntax:

- **`<level>`** is the numeric package level.

The command line switches (/v, /s, /w, and /x) are processed directly by the distribution file to control behavior of the Windows Setup application.

The command line parameters are passed to the Windows Installer (**msiexec**) to control the extracted Windows installer package file (.msi) behavior during the install process.

Command Line Switches

The following table describes the command line switches available for a command line installation:

/v	Passes parameters to the Windows Installer (**msiexec**). The list of parameters must be enclosed in double (") quotation marks. See Command Line Parameters for available parameters.
/s	Suppresses the initialization and extraction dialogs displayed before the product install Welcome dialog." If you are using the #/q command line parameter, use this switch additionally for a completely silent install.

/w	Instructs the Windows Setup application to wait until the installation completes. Use this switch when launching the installation from a script file. Without it, the Setup application may return immediately after launching Windows Installer.
/x	Uninstalls Universal Agent for SOA for Windows.

Command Line Parameters

The following table describes the parameters that are available for a command line installation.

The parameters can be specified in any order, with the following exceptions:

- If the Repair (/fom) or Remove (/x) parameter is used, it must be specified ***before*** all other parameters.
- If the Silent install (/q) and/or Log file (/L) parameters are used, they can be specified in any order, but they must be specified ***after*** all other parameters.

These parameters are preceded by the /v command line switch and must be enclosed in double (") quotation marks.

Parameter	Description	Default
/fom	Repairs a Universal Agent for SOA for Windows installation. **om** (after the **/f**) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the **om** options must be used. **/fom** cannot be used with the **/x** (remove) parameter.	n/a
/x	Removes a Universal Agent for SOA for Windows installation. **/x** cannot be used with the **/fom** (repair) parameter.	n/a
INSTALLDIR =*installdir*	Sets the root installation directory to <**`installdir`**>. All components are installed in this directory. **INSTALLDIR** is required if you want to install Universal Agent for SOA for Windows under a directory different from the one specified by the **PROGRAMFILES** environment variable (typically **`C:\Program Files\Universal`**). If the directory contains spaces, you must use double (**"**) quotation marks around the path name.	(none)
/q	Suppresses the product installation dialogs. Use this parameter in addition to the /s command line switch for a completely silent install. See Command Line Switches, Command Line Installation Examples, and Detecting the Completion of a Silent Install for additional information regarding silent installs.	n/a
/L*v	Instructs the installation process to create an installation log file named <**logfilepath**> (full path name). If <**logfilepath**> contains spaces, you must enclose it with double (") quotation marks. ***v** are flags used to specify the level of detail (**verbose**) contained in the log file. To reduce the amount of output generated, ***v** can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.	n/a

Command Line Installation Examples

The following examples illustrate different ways that Universal Agent for SOA for Windows can be installed from the command line.

Graphical User Interface Install, All Components

To install all Universal Agent for SOA components via the graphical user interface, issue the following command:

```
sb-soa-6.7.0.x-windows-x64.exe
```

Silent Install

To silently install all Universal Agent for SOA components, issue the following command:

```
sb-soa-6.7.0.x-windows-x64.exe /s /v"/qn"
```

Graphical User Interface Install, All Components, with Log File

To install all Universal Agent for SOA components using the Windows Installer graphical user interface and write a log file to `C:\temp\install.log` during the installation, issue the following command:

```
sb-soa-6.7.0.x-windows-x64.exe /v"/l*v C:\Temp\install.log"
```

Silent Install, Different Installation Directory

To silently install all Universal Agent for SOA components to a non-default location, issue the following command:

```
sb-soa-6.7.0.x-windows-x64.exe /s /v"/qn INSTALLDIR=D:\Universal"
```

Detecting the Completion of a Silent Install

If the **/q** switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that can be used to detect when the Windows Installer process ends is to execute it using the system's **start** command. Using available command line switches, **start** can initiate the Windows Installer process and then wait for it to finish. When **start** returns control to its calling process (for example, the command prompt), the process will have ended.

For example, from the command prompt, issue the following command to start the Universal Agent for SOA installation and wait for it to finish:

```
start /b /wait sb-soa-6.7.0.x-windows-x64.exe /w /s /v"/qn"
```

- The **/b** switch prevents the start command from opening a new window.
- The **/wait** parameter causes the start command to start the application, sb-soa-6.7.0.x-windows-x64.exe, and then wait for it to finish.

You also can use the syntax above to execute the **start** command from within a script, such as a **.bat** file.

For more information on the **start** command, open a Windows command prompt and enter: **start /?**.

Modifying a SOA for Windows Installation via the Command Line

- Modifying a Universal Agent for SOA Installation via the Windows Installer Command Line Interface
- Repairing a Corrupted Universal Agent for SOA Installation
  - Silent Repair
  - Interactive Repair, with Log File
- Removing a Universal Agent for SOA Installation
  - Silent Removal

Modifying a Universal Agent for SOA Installation via the Windows Installer Command Line Interface

This page describes how to modify a Universal Agent for SOA installation via the Windows Installer command line interface.

After installing Universal Agent for SOA, you may rerun the installation program as needed in order to:

- Repair a Corrupted Universal Agent for SOA Installation.
- Remove a Universal Agent for SOA Installation.

(For a description of the parameters used in these procedures, see Windows Installer Command Line Parameters.)

Repairing a Corrupted Universal Agent for SOA Installation

To recover accidentally deleted application files, configuration and component definition files, or registry entries required by Universal Agent for SOA using the Windows Installer command line interface, use the **/f** switch together with the **om** parameters.

These are the same repair options set internally by the graphical interface installation. They cause the installation program to reinstall files that are missing or are older than the version contained in the Universal Agent for SOA distribution file.

Silent Repair

To repair a Universal Agent for SOA installation from the command line, issue the following command:

```
msiexec.exe /fom "SetupPath\Stonebranch Universal Agent for SOA (x64).msi" /q
```

Interactive Repair, with Log File

To repair a Universal Agent for SOA installation interactively while writing a log file to **C:\Temp\repair.log**, issue the following command:

```
msiexec.exe /fom "SetupPath\Stonebranch Universal Agent for SOA (x64).msi" /L C:\Temp\repair.log
```

Removing a Universal Agent for SOA Installation

To uninstall a Universal Agent for SOA installation using the Windows Installer command line interface, use the **/x** switch. Issue the following command:

Silent Removal

To uninstall a Universal Agent for SOA installation without using the Windows Installer command line interface, issue the following command:

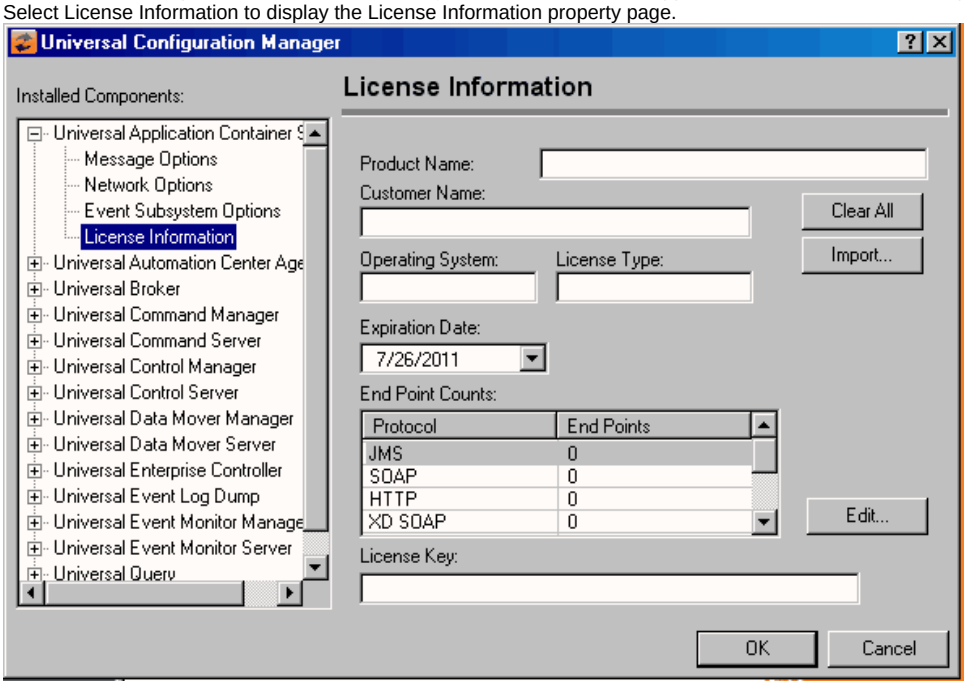
```
msiexec.exe /x "SetupPath\Stonebranch Universal Agent for SOA (x64)".msi /q
```

Universal Agent for SOA for Windows - Configuring and Starting UAC Server

Universal Agent for SOA for Windows - Configuring and Starting the UAC Server

After installing Universal Agent for SOA for Windows, perform the following steps before attempting to configure and start the Universal Application Container Server.

(For more information, see the UAC Server section in the Universal Command Agent for SOA 7.2.x Reference Guide.)

Step 1	Refresh the local Universal Broker's cached configuration. Starting with version 3.2.0.0, Universal Broker caches configuration and component definition information for all Universal Agent applications. Before UAC Server will run, its configuration and component definition information must reside in the local Broker's cache. To refresh the local Broker's cache without stopping and restarting the Broker, issue a REFRESH command via Universal Control (For more information, see the Universal Agent 7.2.x User Guide.)
Step 2	Add Universal Application Container (UAC) Server license information to the UAC Server configuration file. Stonebranch, Inc. provides UAC Server license information in a formatted text file. This file contains the required license parameter keywords and values. You can simply append this license information to the UAC Server configuration file, **uacs.conf**, or import it using the Universal Configuration Manager. To import license information using the Universal Configuration Manager. 1. Open the Windows Control Panel to start the Universal Configuration Manager. 2. In the Installed Components tree, click on the **+** symbol next to Universal Application Container Server to display its property pages. 3. Select License Information to display the License Information property page.  4. If you want to enter the license information manually on this page, make sure that you enter the information exactly as it appears in the license information file. • All information is case\-sensitive. • You must enter all punctuation and spacing as provided in the file. Instead of entering the information manually, you can import the license file by clicking the **Import** button, and then opening the file from the Select License File dialog. 5. After entering or importing the license information, click the **OK** button to save the information in the UAC Server configuration file.
Step 3	Recycle the local Universal Broker to start the UAC Server.

By default, the UAC Server is set to start automatically whenever its local Universal Broker starts. Stopping and restarting the Universal Broker will cause UAC Server to start.

Universal Agent for SOA for Windows - 64-Bit Windows Editions

Universal Agent for SOA for Windows - Installing on 64-bit Windows Editions

All Universal Agent for SOA components have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows 7
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

The installation defaults for Universal Agent for SOA should not require any modification when installing on 64-bit Windows editions.

Universal Agent for SOA for Windows - File Inventory Lists

- Universal Agent for SOA for Windows - File Inventory Lists
- Universal Application Container Server
- Universal Application Container
- Universal Application Interface
- OpenJDK Runtime Environment, 1.8.0_212-3-redhat-b04
- System Files

Universal Agent for SOA for Windows - File Inventory Lists

The Universal Agent for SOA 6.3.x for Windows package includes the files required for the following components / utilities:

- Universal Application Container Server
- Universal Application Container
- Universal Application Interface
- OpenJDK Runtime Environment, 1.8.0_212-3-redhat-b04

This page lists the files installed with each Universal Agent for SOA component. The file paths specified are relative to the root installation directory (for example, **C:\Program Files\Universal**) that was specified during the installation.

Universal Application Container Server

File	Description
UAC\startUAC.bat	Starts the Universal Application Container.
UAC\shutdownUAC.bat	Stops the Universal Application Container.
UAC\UAC.xsd	File used by Universal Application Container to validate content of the **UAC.xml** inbound configuration file. It requires no user interaction.
UAC\derby.log	Universal Application Container database log file.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uac_log4jConfiguration.xml	Universal Application Container logging configuration file that you may need to modify if you want to change the logging level of the UAC component.
UAC\uacValidateInbound.bat	Script that validates the contents of the **UAC.xml** file using the schema definition contained in **UAC.xsd**.
UAC\bin\uacsrv.exe	Universal Application Container Server program file.
UAC\bin\uacscfg.dll	Used by Universal Configuration Manager to manage Universal Application Container Server configuration options.
UAC\lib\uacsrv.jar	Universal Application Container Server class file.
UBroker\tmpl\uaccfg	Universal Application Container Server configuration template file.
UBroker\tmpl\uaccmp	Universal Application Container Server component definition template file.
nls\uacmceng.umc	Universal Application Container Server message catalog.
%ProgramData%\Universal\conf\UAC.xml	Inbound operation configuration file that you may need to modify to support your inbound workload operations. **%ProgramData%**, by default, resolves to C:\ProgramData.
%ProgramData%\Universal\conf\uacs.conf	Universal Application Container Server configuration file. **%ProgramData%**, by default, resolves to C:\ProgramData.
%ProgramData%\Universal\comp\uac	Universal Application Container Server component definition file. **%ProgramData%**, by default, resolves to C:\ProgramData

Universal Application Container

File	Description
UAC\container	Sub-directory containing the libraries and other deployable objects needed for UAC operation.

Universal Application Interface

File	Description
UAI\lib	Sub-directory containing the **uacsrv.jar** file needed for communication between the **uacsrv** and **uac** components. It requires no user interaction.
UAI\.keystore	File that contains the certificates for secure communication between UAC and UAI.
%Program Data% \Universal\conf\uai_log4jConfiguration.xml	Universal Application Interface logging configuration file. **%ProgramData%**, by default, resolves to C:\ProgramData.
UAI\uai.bat	Universal Application Interface start-up script.

OpenJDK Runtime Environment, 1.8.0_212-3-redhat-b04

File	Description
jre*.*	Files required for redistribution of the OpenJDK Runtime Environment.

System Files

The following files will be installed only if they are newer than the existing file.

All files are installed to the Windows System32 folder, unless otherwise noted.

Description	File
Visual C++ 10.0 Runtimes	msvcp100.dll msvcr100.dll
Visual C++ 14 Runtimes	concrt140.dll msvcp140.dll msvcp140_1.dll msvcp140_2.dll ucrtbase.dll vccorlib140.dll vcruntime140_1.dll
Microsoft Foundation Class 14.0	mfc140.dll mfc140u.dll mfcm140.dll mfcm140u.dll mfc140chs.dll mfc140cht.dll mfc140deu.dll mfc140enu.dll mfc140esn.dll mfc140fra.dll mfc140ita.dll mfc140jpn.dll mfc140kor.dll mfc140rus.dll

Microsoft Universal C Runtime [1]	api_ms_win_core_console_l1_1_0.dll api_ms_win_core_datetime_l1_1_0.dll api_ms_win_core_debug_l1_1_0.dll api_ms_win_core_errorhandling_l1_1_0.dll api_ms_win_core_file_l1_1_0.dll api_ms_win_core_file_l1_2_0.dll api_ms_win_core_file_l2_1_0.dll api_ms_win_core_handle_l1_1_0.dll api_ms_win_core_heap_l1_1_0.dll api_ms_win_core_interlocked_l1_1_0.dll api_ms_win_core_libraryloader_l1_1_0.dll api_ms_win_core_localization_l1_2_0.dll api_ms_win_core_memory_l1_1_0.dll api_ms_win_core_namedpipe_l1_1_0.dll api_ms_win_core_processenvironment_l1_1_0.dll api_ms_win_core_processthreads_l1_1_0.dll api_ms_win_core_processthreads_l1_1_1.dll api_ms_win_core_profile_l1_1_0.dll api_ms_win_core_rtlsupport_l1_1_0.dll api_ms_win_core_string_l1_1_0.dll api_ms_win_core_synch_l1_1_0.dll api_ms_win_core_synch_l1_2_0.dll api_ms_win_core_sysinfo_l1_1_0.dll api_ms_win_core_timezone_l1_1_0.dll api_ms_win_core_util_l1_1_0.dll api_ms_win_crt_conio_l1_1_0.dll api_ms_win_crt_convert_l1_1_0.dll api_ms_win_crt_environment_l1_1_0.dll api_ms_win_crt_filesystem_l1_1_0.dll api_ms_win_crt_heap_l1_1_0.dll api_ms_win_crt_locale_l1_1_0.dll api_ms_win_crt_math_l1_1_0.dll api_ms_win_crt_multibyte_l1_1_0.dll api_ms_win_crt_private_l1_1_0.dll api_ms_win_crt_process_l1_1_0.dll api_ms_win_crt_runtime_l1_1_0.dll api_ms_win_crt_stdio_l1_1_0.dll api_ms_win_crt_string_l1_1_0.dll api_ms_win_crt_time_l1_1_0.dll api_ms_win_crt_utility_l1_1_0.dll ucrtbase.dll
Microsoft Windows Installer 3.1.4000.1823 [2]	msiexec.exe

[1] The Microsoft Universal C Runtime is built into Windows 10, Server 2016, and Server 2019. It is available as a Microsoft update on Windows 7, Server 2008 R2, Server 2012, and Server 2012 R2 systems. Although the Universal Agent install provides these .dlls on older system, the install will issue an error and stop without installing any files if they do not already exist. The install executes functions that rely on the MS UCRT.

[2] See the documentation regarding the Microsoft Windows Installer.

Windows Installation - Licensing

- Licensing Universal Agent for Windows Components
- Product License File
  - Format
  - Sample
- Entering License Information
  - Licensing via Universal Configuration Manager
- Restart Universal Broker
  - Via Windows Services
  - Via Universal Configuration Manager

Licensing Universal Agent for Windows Components

After Universal Agent for Windows has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Connector for SAP
- Universal Connector for PeopleSoft
- Universal Application Container Server

Note

Universal Application Container Server (as a component of Universal Command Agent for SOA 6.7.x for Windows) is packaged, and licensed, separately.

Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

Format

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: **Indesca_Stonebranch_NT_A1.txt**

- For Universal Command Manager, **Indesca** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.
- For Universal Data Mover Manager, **Infitran** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.

Sample

The following is a sample Universal Command Manager for Windows product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "NT"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Entering License Information

Universal Agent for Windows components can be configured with product licenses either by:

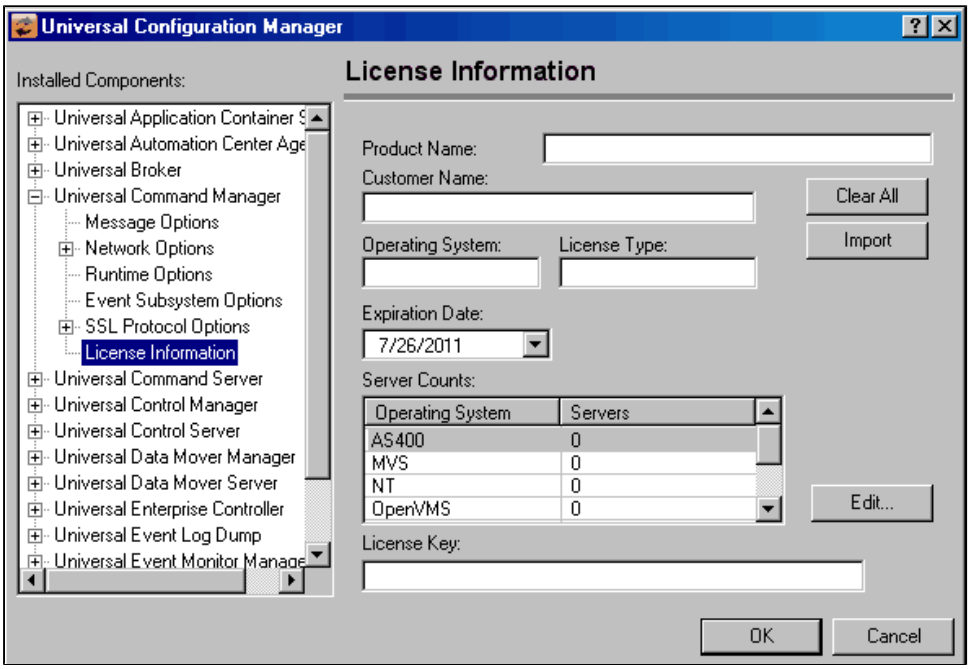
1. Entering the information from their product license file into their configuration file. It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)
    - Universal Command Manager: **ucmd.conf**
    - Universal Data Mover Manager: **udm.conf**
    - Universal Connector for SAP: **usap.conf**
    - Universal Connector for PeopleSoft: **upps.conf**
    - Universal Application Container Server: **uacs.conf**
2. Specifying the information on the Universal Configuration Manager License Information page for that component, either by.
    a. Entering the information specified in the license file.
    b. Importing the license file.

Note

Universal Configuration Manager is installed during the Universal Agent 7.2.x for Windows installation.

Licensing via Universal Configuration Manager

To enter license information via the Universal Configuration Manager, perform the following steps:

Step 1	On the Windows Control Panel, double-click the Universal Configuration Manager icon to display the Universal Configuration Manager screen.
Step 2	In the Installed Components tree, click **+** next to the component that you want to configure with license information.
Step 3	Click License Information to display the License Information page. 
Step 4	If you want to enter the license information on the page, make sure that you enter the information exactly as it is in the license information file. All information is case-sensitive; punctuation and spacing must be maintained.
Step 5	If you want to import the license file: 1. Click the Import button. 2. On the Select License File dialog that displays, select the license file to import. 3. Click the Open button. The information is imported to the License information page.
Step 6	Click the OK button to save the license information.

Restart Universal Broker

For Universal Broker to read the license information, you must start / restart it.

Via Windows Services

Step 1	Go to Windows Services.
Step 2	Locate and select Universal Broker.
Step 3	If Universal Broker is not running, click **Start**. If Universal Broker is running, click **Restart**.

Via Universal Configuration Manager

If you enter license information via the Universal Configuration Manager (see above), Universal Broker is restarted automatically, and the license information is read, when you click OK to save the license information.

UNIX Installation

- Introduction
- Installation Packages
- Universal Agent Installation Methods
  - System Installation
  - User Mode Installation
- Universal Agent Installation Summary
- Detailed Information

Introduction

These pages provide information on the installation of Stonebranch, Inc.'s Universal Agent on UNIX operating systems. Unless otherwise specified, all references to Universal Agent for UNIX refer to version 7.0.x.

Note

This information does not include installation on z/OS UNIX System Services (USS). See z/OS USS Installation for z/OS USS installation instructions.

Installation Packages

Universal Agent is installed with one required installation package and one optional installation package:

Package	Distribution File	Required
Universal Agent for UNIX Installation	(See Universal Agent for UNIX - Distribution File.)	Yes
Universal Agent for SOA for UNIX Installation	(See Universal Agent for SOA for UNIX - Distribution File.)	No

Note

Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Agent component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

Universal Agent Installation Methods

There are two installation methods supported on UNIX platforms for installing the Universal Agent package: system installation and user mode installation.

For detailed information on each method, see the platform-specific installation instructions:

- AIX installation
- HP-UX installation
- Solaris installation
- Linux installation

System Installation

The system installation method uses UNIX operating system vendor installation tools to install Universal Agent in the vendor-recommended installation directories.

This is the Stonebranch legacy method for installing UNIX Agents.

User Mode Installation

The Stonebranch install script, `unvinst`, offers a `-usermode_install` parameter that permits multiple Universal Agent installs on a single system. Additionally, a user mode install can create an environment in which an Agent executes as an unprivileged user (with significantly reduced functionality).

The user executing the install determines the environment in which the Agent executes. The root account may create the following environments:

- A fully-privileged system install to default locations.
- An "elevated" user mode install to custom locations. This is an install executed with the `-usermode_install` parameter, and permits user-specified install directories with some program execution under root authority.

When a non-root user executes the install script, the `-usermode_install` parameter is required. The result of an unprivileged install is one in which all application files are owned and execute as the (unprivileged) Broker user.

An unprivileged user mode environment will impose significant restrictions on Agent functionality, but may be desirable in test situations or for executing light workload.

The following requirements must be met in order to install and execute an Agent in an unprivileged user mode environment:

1. The Broker user must exist prior to the install. The account must have an existing home directory.
2. The account performing the install does not need to be root or the Broker user, but it must have authority to write to the Broker user's home directory.
3. A user mode install must be done by passing the `-usermode_install` parameter to unvinst.
4. The "security" configuration option must be set to "none" in `uags.conf`, `ucmds.conf`, `uctls.conf`, `udms.conf`, and `uems.conf`. This means that no user context switching is supported in an unprivileged user mode environment. All tasks execute as the Broker user.
5. Any task submitted via the Controller to execute on the Agent must not have credentials specified. This is necessary to ensure the process is spawned without requiring any kind of root privileges. One exception to this condition applies to any task that supports the "Run as sudo" option. In that case, credentials may be supplied, but must match an existing sudoers entry.
6. Any user that wants to execute Agent Managers (for example, ucmd, udm, or uem) or Utilities (for example, uctl or uquery) must source the `SB_ConfigSetup.env` file within their user profile. This file is installed in the `./etc` directory located under the install directory, and sets environment variables that point to configuration files and to the Broker's `.bif` and `.plf` files.

Universal Agent Installation Summary

Step 1	Download the distribution file from the Stonebranch Customer Portal.
Step 2	Copy the distribution file to the UNIX system.
Step 3	Logon to the UNIX system as root or execute **su** (substitute user) command to root.
Step 4	Uncompress the distribution file.
Step 5	Extract the files from the uncompressed **tar** file.
Step 6	Run the Stonebranch, Inc. installation script, **`unvinst`**.
Step 7	Run the Universal Broker daemon start-up script (located in **`./universal/ubroker`**) to start the Universal Broker: **`ubrokerd start`**.

Detailed Information

The following pages provide detailed information for UNIX Installation:

- Universal Agent for UNIX Installation
- Universal Agent for SOA for UNIX Installation
- Licensing

Universal Agent for UNIX Installation

Overview

The following information is provided for the installation of Universal Agent for UNIX:

- Installation Package
- Installation Requirements
- Distribution File
- Installation Procedures
- Customization
- File Inventory Lists

(For licensing information, see UNIX Installation - Licensing.)

Universal Agent for UNIX - Installation Package

- Components
- Component Compatibility
- Licensing

Components

The Universal Agent 7.2.x for UNIX package includes the following product components:

- Universal Broker 7.2.x
- Universal Automation Center Agent 7.2.x
- Universal Certificate 7.2.x
- Universal Command Manager and Server 7.2.x
- Universal Connector for SAP 7.2.x
- Universal Connector for PeopleSoft 7.2.x (for Linux)
- Universal Control Manager and Server 7.2.x
- Universal Data Mover Manager and Server 7.2.x
- Universal Encrypt 7.2.x
- Universal Event Monitor Manager and Server 7.2.x
- Universal Message to Exit Code Translator 7.2.x
- Universal Query 7.2.x
- Universal Message Service (OMS) 7.2.x
- Universal Controller Command Line Interface (CLI) 7.2.x
- Python 3.7 Distribution for Universal Agent (optional) (not yet available for HP-UX)

Component Compatibility

The following table identifies the compatibility of Universal Agent 7.2.x for UNIX components with previous component / product versions.

Component	Compatibility
Universal Broker 7.2.x	Universal Agent / Universal Products releases 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Command 7.2.x	Universal Command 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 7.2.x	Universal Control 7.1.0.0, 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 7.2.x	Universal Data Mover 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1 and 3.1.0.
Universal Encrypt 7.2.x	Universal Encrypt 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 7.2.x	Universal Broker 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Event Monitor 7.2.x	Universal Event Monitor 7.1.0.0, 7.0.0.0, 6.9.0, 6.8.0, 6.7.1, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, and 3.1.0.

The component references pertain to all supported UNIX platforms for that version.

Licensing

For licensing information, see UNIX Installation - Licensing.

Universal Agent for UNIX - Installation Requirements

- Installation Requirements
  - UNIX Versions
  - Additional Requirements
  - Platform Requirements
- Directories and Files
  - Command Reference Directory
  - Log Directory
  - Trace Directory
  - Spool Directory
  - UAG Cache

Installation Requirements

In order to install Universal Agent components, you must be able to write to the directory from which the installation is launched.

UNIX Versions

One of the following UNIX operating systems that are supported by Universal Agent 7.2.x:

- AIX 6.1 TL1 and above, 64-bit
- HP-UX 11.23 and above (HP-UX IA64 package)
- Solaris 10 and above (SPARC-based)

Note

Universal Agent for Solaris 10 systems require libxnet.so version SUNW_1.3.1 or higher.

Note

The Python Distribution for Universal Agent requires libintl.so version 0.18.1.1, REV=2011.03.15 or higher. The install will proceed if the required dependency does not exist, but the UA-specific python instance will not run until libintl is available.

- Solaris 10 and above (Intel-based)

Note

The Python Distribution for Universal Agent requires libintl.so version 0.18.1.1, REV=2011.03.15 or higher. The install will proceed if the required dependency does not exist, but the UA-specific python instance will not run until libintl is available.

- Linux Environments with the following qualifications:
  - Intel x86_64-based system
    - Linux Kernel 3.10 or greater
    - glibc 2.17-78 or higher
  - IBM PowerPC-based system
    - Linux Kernel 3.10 or greater
    - glibc 2.17-157 or higher
    - Configured for little endian (LE)
  - zSeries 64-bit (S/390) system
    - Linux Kernel 2.6 or greater
    - glibc 2.4-31 or greater
  - RedHat Package Manager (RPM) distributions
  - Debian distributions
    - Linux Kernel 3.0 or greater
    - glibc 2.15 or higher

The list of supported systems grows rapidly. Contact Stonebranch, Inc. for a current list of supported UNIX operating systems if you require support for a system that is not listed.

Additional Requirements

- Superuser (root) access.
- TCP/IP Socket implementation.
- Approximately 400 megabytes of disk space for the installation. More disk space is required for variable files, such as log files, spool files, and trace files.
- Bourne shell or compatible.

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x to make sure that your platform is supported before performing an installation.

Directories and Files

All product files that are written to during product execution are stored in the **/var/opt/universal** directory by default. This section documents the estimated amount of space required on the file system for all directories (and their sub-directories), required security access, and mount requirements.

Command Reference Directory

Universal Command Server can execute commands of type **cmdref**. A command reference is a predefined command or script to which the Universal Command Manager refers by its file name.

The command reference directory is **/var/opt/universal/cmdref**.

Space

The amount of space required is solely dependent on the number of command reference files you define. No command reference files are included in the installation.

Security

Universal Command Server requires read access to the **cmdref** directory. The product administrator requires read/write access in order to maintain the command reference files. No general user access is required.

Log Directory

Universal Broker can be configured to write its messages to a log file.

The current log file and previous log file generations are stored in the **/var/opt/universal/log** directory.

Space

A log file size grows to about 150,000 bytes and is then rolled over to a generation file. Five generations of log files are saved. The oldest generation log file is deleted. The amount of space required for the five generations and the current log file is about 900,000 bytes.

Security

Universal Broker requires read/write access to the log directory and read/write access to all files in the log directory. No other Universal Agent components use the log directory at this time. No general user access is required.

Trace Directory

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. Customer Support to resolve product problems.

Space

Trace files can grow to significant size depending on how long the trace is active and how much work the program is doing during the tracing period. A trace file size of about 10MB is not unusual.

Trace file sizes can be limited by setting the maximum number of lines to write to a file with the MAX_TRACE_LINES configuration option. Once the maximum is reached, the trace file will wrap to the beginning.

Under normal operation, no space is required for trace files. Trace files are requested by Stonebranch, Inc. Customer Support only for problem resolution. When trace files are required, at least 20MB of disk space should be available.

Security

Universal Broker and the Broker components (Universal Command Server and Universal Control Server) require read/write access to the trace directory. No other Universal Agent components access the trace directory. No general user access is required.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Agent components started by Universal Broker.

- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with Manager Fault Tolerance enabled.

The default location for the spooled standard I/O files and other database files is the **/var/opt/universal/spool** directory.

Universal Command Server

Space

Spool files are located, by default, in directory **/var/opt/universal/spool**. The location of various product files can be can be configured via the product configuration files.

The amount of disk space required for the spool directory depends on:

1. Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
   When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
2. Average size of the user processes standard input, output, and error files.

A Stonebranch Tip

Keep in mind that spooling is not a feature for file transfer purposes.

File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

```
MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = required disk space.
```

For example:

If the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. The feature must be turned on through the ALLOW_SPOOLING configuration settings.

For more information on the Manager Fault Tolerant feature and the spooling of redirected standard I/O files, see the Universal Agent 7.2.x User Guide.

Security

Universal Broker, Universal Command Server and Universal Event Monitor require read/write access to the spool directory. No other Universal Agent components access the spool directory. No general user access is required.

Mount

The spool directory must be mounted on a local device. It cannot be mounted on a network device, such as an NFS or SAMBA mount.

UAG Cache

UAG cache is used by Universal Automation Center Agent (UAG) for storing standard I/O files.

Space

Cache files are located, by default, in directory **`/var/opt/universal/uag/cache`**.

Cache files are created for each job that is run by the agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by Universal Controller. You can configure the number of days that files remain in the cache via the Universal Controller user interface (see the Agent Cache Retention Period in Days Universal Controller system property).

The amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Automation Center Core daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

```
(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.
```

For example:

If the files are purged every 7 days, and you run 1200 jobs on that agent server daily, and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

The Universal Automation Center Agent Server automatically redirects the standard output and standard error files to the cache directory with no required user input.

Security

Universal Automation Center Agent require read/write access to the UAG cache directory. No other Universal Agent components access the cache directory. No general user access is required.

Universal Agent for UNIX - Distribution File

- UNIX Distribution File
- Obtaining the Distribution File
- Distribution File Format

UNIX Distribution File

Stonebranch, Inc. provides different Universal Agent for UNIX packages for different types of UNIX operating systems.

Obtaining the Distribution File

To obtain the Universal Agent for UNIX package for your type of UNIX operating system, you must download the corresponding product distribution file from the Stonebranch Customer Portal.

Note

A customer user name and password — provided by Stonebranch, Inc. — are required to access this area.

After a distribution file has been downloaded, the installation files contained in that file must be extracted before the product can be installed (see Universal Agent for UNIX - Installation Procedures).

Distribution File Format

The name of each Universal Agent for UNIX distribution file has the following format:

`sb-Version.Release.Modification Level.Maintenance Level-operating system-version(.release)(-platform).tar.Z`

For example: **`sb-7.2.0.0-linux-2.6-x86_64.tar.Z`**

In this format:

- **Version** is the current version of Universal Agent.
- **Release** is the current release of Universal Agent.
- **Modification Level** is the current Universal Agent feature set.
- **Maintenance Level** is the Universal Agent build level.
- **operating system** is the name of the operating system (for example, AIX or Linux).
- **version(.release)** is the supported version and, optionally, the release of the operating system.
- **platform** is the targeted hardware platform (for example, RS6000 or i386). It is included in the file name only if there is more than one platform available for the specified operating system.

Universal Agent for UNIX - Installation Procedures

Universal Agent for UNIX Installation Procedures

The following procedures are provided for the installation and modification of Universal Agent for UNIX:

- Universal Agent for AIX Installation
- Universal Agent for HP-UX Installation
- Universal Agent for Solaris Installation
- Universal Agent for Linux Installation

Universal Agent for AIX Installation

- Installation Process
- Extracting the Universal Agent for AIX Installation Files
  - Distribution File
- Installing Universal Agent for AIX
  - Component Selection
  - Starting the Installation Script
  - AIX Installation Script Parameters
  - Installation Script Example
  - User Mode Installation
  - AIX PAM Customization
- Listing Universal Agent for AIX Information
- Removing Universal Agent for AIX
  - System Install Removal
  - User Mode Install Removal

Installation Process

Installation of Universal Agent for AIX is a three-step process:

Step 1	Download the product distribution file.
Step 2	Extract the installation files from the distribution file.
Step 3	Install the extracted files.

Extracting the Universal Agent for AIX Installation Files

The Universal Agent for AIX product distribution file is in a compressed **tar** format.

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat sb-7.2.x.x-aix-7.1-ppc64.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is **`sb-7.2.x.x-aix-7.1-ppc64.tar.Z`**.
- File is located in the current working directory.

Distribution File

The distribution file contains multiple files, including packages in the AIX backup file format, using extension **.bff**. The actual base name of the **.bff** file depends on the AIX version for which the distribution file is intended. (See Distribution File Format for distribution file naming conventions.)

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv-7.2.x.x-aix-7.1-ppc64.rpm	AIX RPM file format package.
unv-opscli-7.2.x.x-aix-7.1-ppc64.rpm	Universal Controller Command Line Interface (CLI) RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.

usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for AIX distribution file does not contain all of these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

Installing Universal Agent for AIX

Universal Agent for AIX is installed with the **`unvinst`** script, which executes the **rpm** command. The command to start the script must be executed with the superuser ID.

Note

Stonebranch, Inc. strongly recommends the use of the **`unvinst`** script for the AIX installation above any other method.

Component Selection

The Universal Agent package contains many components, which are grouped into five categories. Components in some categories are installed and activated optionally, as specified in the **`unvinst`** script by using **`unvinst`** command line parameters.

The following table describes each category, provides the default installation configuration for the components in that category, and identifies the command line parameters to use for components that are optionally installed and activated.

Note

The default installation configuration refers to new installs only. For upgrades, installed component configurations are not changed by the upgrade process.

Category	Default Configuration	Description
Base components	Always installed.	Base components are always installed and activated. They include the Universal Broker, Universal Command (UCM), Universal Data Mover (UDM), and others. Base components provide the core agent infrastructure and workload services.
Universal Automation Center Agent (UAG)	Always installed, but inactive.	Universal Controller utilizes UAG agents to provided distributed, workload automation services. UAG is activated using the --oms_servers parameter.
Universal Message Service (OMS)	Always installed, but inactive.	OMS is message-oriented middleware that should be deployed on a small set of centrally located servers. It should not be activated on every Agent install. OMS is the network provider for UAG and the Universal Controller. OMS is activated using the --oms_autostart parameter.
Universal Controller command line programs	Not installed.	Universal Controller command line programs provide a command line interface to the Universal Controller. The installation of Universal Controller command line programs is optional. The command line programs are only required on Agents that need to interface with the Universal Controller via the command line. Universal Controller command line programs are installed using the --opscli parameter.
Third-party components	Not installed.	This category includes components not developed or maintained by Stonebranch, Inc. For example, Universal Agent 7.2.0.0 provides a Python 3.7 distribution that can be installed under the Universal Agent install directory. This distribution is provided so that a reliable Python environment exists on systems that may not provide for all users.

Starting the Installation Script

To start the **`unvinst`** installation script and install all of the AIX packages, issue the following command:

```
sh ./unvinst [--user username [--userdir directory] [--create_user {yes|no}] ] [--group group name [--
create_group {yes|no}] ]
             [--keystore {yes|no}] [--convert_opsagent [--opsdir directory] ] [--oms_servers network address]
[--oms_port port]
             [--ac_agent_clusters clusters] [--ac_agent_ip IP address] [--oms_autostart {yes|no}] [--ac_netname
ID]
             [--ac_extension_accept_list list] [--ac_extension_python_list list] [--
ac_extension_deploy_on_registration {yes|no}]
             [--ac_extension_cancel_timeout value {s|m|h|d}] [--ac_process_cancel_timeout value {s|m|h|d}]
             [--uag_autostart yes] [--opscli {yes|no}] [--usermode_install {yes|no} [--unvdir directory]
             [--unvcfgdir directory] [--unvdatadir directory] [--unvport port] [--python {yes|no}]  ]
```

See AIX Installation Script Parameters and Installation Script Example, below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

Note

If you want to install multiple Agents on the same machine, or add one or more additional Agents to a machine with a previously installed Agent, some of these installation parameters are required (see Installing Multiple Agents on a Single Machine).

This is a silent install. The output from **unvinst** is written to file **install.log** in the current directory.

An entry is added to the system initialization table, **/etc/inittab**, to start the **ubrokerd** daemon when the system boots to runlevel 2. The **inittab** entry is similar to the following:

```
ubroker:2:once:/opt/universal/ubroker-7.2.0/ubrokerd start
```

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the installation script parameters, below.

Note

With the Solaris, HP, and AIX installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain and can be reviewed and removed, as desired, by your Administrator.

AIX Installation Script Parameters

The following table describes the optional parameters that are available in the installation script, **unvinst**, when installing Universal Agent.

The parameters are grouped into the following categories:

Category	Description
Base	Used for base install.
OMS	Used for an active OMS configuration.
UAG	Used for an active UAG configuration.
CLI	Used for Universal Controller CLI programs install.
User Mode	Used for user mode installation.
Third-party	Used to manage third-party components.

Parameter	Description	Default
Base Parameters		

-u **user** **--user**	Normal UNIX **username** that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Universal Agent server components (for example: **ucmsrv**, **udmsrv**, and **uemsrv**) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set. • If the user account that you want to use already exists, specify that user account. • If the user account does not exist, the install creates it. • If you want to change the user account for an installed Universal Agent for AIX system, you must perform a re-installation and use this parameter to change the user account. • If **--user** is omitted from **`unvinst`**, the default is used. • If --usermode_install is **yes**, there is no default.	**ubroker**
create_user **create_user**	Specification (**yes** or **no**) for whether or not to create the user name that will own the installed files as a local user.	**yes**
userdir **userdir**	Home directory for the created user account specified by --user. • If this directory does not exist, it is created when the specified user is created. • If the user specified by --user already exists, but the home directory of that user is not the default directory (**`/home/<username>`**), **`--userdir`** must specify the path to that home directory. • If **--userdir** is omitted from **`unvinst`**, the default is used.	**`/home`** **`/<username>`**
-g **group** **group**	Normal UNIX **groupname**; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group. • If the group that you want to use already exists, specify that group. • If this group does not exist, the install creates it. • If **--group** is omitted from **`unvinst`**, the default is used.	**ubroker**
create_group **create_group**	Specification (**yes** or **no**) for whether or not to create the group that will own the installed files as a local group.	**yes**
keystore **keystore**	Specification (**yes** or **no**) for whether or not to create encryption keys during installation and set up the local Universal Broker as a keystore owner.	**no**
UAG Parameters		
-c **convert** **convert_opsagent**	Converts an existing Opswise Automation Center Agent (1.5, 1.6, or 1.7) to a 5.2.0 Agent. The installation process will invoke a script, **`opsmerge.sh`**, which stops the Agent and converts the configuration options stored in its **`agent.props`** file to corresponding configuration options in the Universal Automation Center Agent (UAG) **`uags.conf`** file. The script also performs several other tasks needed for the conversion. Note You must include this parameter when upgrading an Opswise Automation Center Agent (1.5, 1.6, or 1.7) to Universal Agent 7.2.x.x.	
-d **opsdir** **opsdir**	If --convert_opsagent is specified: Identifies the primary install directory for Universal Agent. • If **--opsdir** is omitted from **`unvinst`**, the default is used.	**`/home`** **`/opswise`**
oms_servers **oms_servers**	Specifies a value, in the format **`port@host[,port2@host,...,portn@hostn]`**, for the port and network address of the OMS server(s) to be used as network communications providers. Note You should always include **--oms_servers** in **`unvinst`**; OMS is the network communications provider between Controller 7.2.x.x and Agent 7.2.x.x. **UAG configuration:** The value specified for **--oms_servers** is set automatically for the UAG OMS_SERVERS configuration option.	(none)
ac_agent_clusters **ac_agent_clusters**	Specifies the Universal Controller-defined clusters to which this agent will belong. **UAG configuration:** The value specified by **--ac_agent_clusters** is set automatically for the UAG AGENT_CLUSTERS configuration option.	(none)

ac_agent_ip **ac_agent_ip**	Specifies the IP address or host name (which resolves to an IP address) that the Agent reports to the Controller. **UAG configuration:** The value specified by **--ac_agent_ip** is set automatically for the UAG AGENT_IP configuration option.	(none)
ac_netname **ac_netname**	Specifies the network ID that Universal Agent will use. **UAG configuration:** The value specified by **--ac_netname** is set automatically for the UAG NETNAME configuration option.	(none)
-ac_extension_accept_list **--ac_extension_accept_list**	Specifies a comma-separated list of one or more Universal Extensions that the agent will accept via auto-deployment from the Controller. • A single value of ***** indicates that all extensions are accepted. • A single value of **none** indicates that no extensions are accepted. The value specified for this parameter sets the EXTENSION_ACCEPT_LIST configuration option value.	*****
-ac_extension_python_list --ac_extension_python_list	Specifies a comma-separated list of zero or more Python locations. Each item in the list is expected to contain a complete path to a Python executable. The value specified for this parameter sets the EXTENSION_PYTHON_LIST configuration option value.	/usr/bin/python3,/usr/bin/python,/usr/libexec/platform-python
-ac_extension_deploy_on_registration --ac_extension_deploy_on_registration	Controls Extension deployment behavior from the Universal Controller. • If the value is **yes**, the Controller will preemptively deploy all extensions acceptable by UAG. • If the value is **no**, the Controller will only send Extension modules as needed (on demand). The value specified for this parameter sets the EXTENSION_DEPLOY_ON_REGISTRATION configuration option value.	**no**
-ac_extension_cancel_timeout **--ac_extension_cancel_timeout**	Set the EXTENSION_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time a Universal Extension task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**
ac_enable_ssl **ac_enable_ssl**	*This option is deprecated starting with Universal Agent 7.1.0.0. UAG Server will always attempt to use SSL/TLS for OMS connections.*	**no**
-ac_process_cancel_timeout **--ac_process_cancel_timeout**	Set the PROCESS_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time an OS task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**

uag_autostart **uag_autostart**	Specifies (**yes** or **no**) whether or not the Universal Automation Center Agent (UAG) Server starts automatically when the Universal Broker is started. **UAG component definition:** The value specified by **--uag_autostart** is set automatically for the UAG AUTOMATICALLY_START component definition option.	**yes**
OMS Parameters		
oms_port **oms_port**	Specifies the port to use to listen for OMS connection requests. **OMS configuration:** The value specified by **--oms_port** is set automatically for the OMS SERVICE_PORT configuration option.	(none)
oms_autostart **oms_autostart**	Specifies (**yes** or **no**) whether or not OMS is started automatically by Universal Broker when Universal Broker starts. **OMS component definition:** The value specified by **--oms_autostart** is set automatically for the OMS AUTOMATICALLY_START and RESTART component definition options.	**no**
CLI Parameters		
opscli **opscli**	Specifies (**yes** or **no**) whether or not the Universal Controller Command Line Interface (CLI) tools will be installed.	**no**
User Mode Parameters		
-U **-usermode_install** **--usermode_install**	Specifies (**yes** or **no**) for a user mode installation, which defines both of the following: • Whether or not this Agent can be installed on a machine where an Agent already is installed (see Installing Multiple Agents on a Single Machine). • Whether or not the user can specify installation parameters (--unvdir, --unvcfgdir, --unvdatadir, and --unvport) that control where the Agent will be installed and configured (see Changing the Default Installation Directories).	**no**
-unvdir **--unvdir**	If --usermode_install is set to **yes**: Specifies the Agent binaries (installation) directory.	(none)
unvcfgdir **unvcfgdir**	If --usermode_install is set to **yes**: Specifies the Agent configuration files directory.	**<--unvdir> /etc**
unvdatadir **unvdatadir**	If --usermode_install is set to **yes**: Specifies the Agent data files directory.	**<--unvdir> /var**
unvport **unvport**	If --usermode_install is set to **yes**: Specifies the Universal Broker port.	(none)
Third-Party Parameters		
python **python**	Specifies (**yes** or **no**) whether the Python 3.7 Distribution for Universal Agent is installed.	**no**
Additional Parameter		
-? **-h** **help** **info**	Displays command line help.	n/a

Installation Script Example

The following example illustrates Universal Agent for AIX installed with the installation script, **unvinst**, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup
             --keystore no --convert_opsagent --opsdir /homedir/ops
             --oms_servers 7878@oms2 --oms_port 7878 --oms_autostart yes
             --ac_agent_clusters GA Cluster,CA Cluster --ac_agent_ip 127.0.0.1
             --ac_netname OPSAUTOCONF --opscli yes --uag_autostart yes
             --usermode_install yes --unvdir /opt/universal
             --unvcfgdir /etc/universal --unvdatadir /var/opt/universal --unvport 7887
```

User Mode Installation

A user mode installation, implemented through use of the usermode_install installation parameter, lets you install multiple Agents on a single machine and change the default installation directories for any Agent being installed.

You must perform a user mode installation for installing an Agent on a machine where one or more Agents already have been installed.

However, you also can perform a user mode installation for the initial installation of an Agent on a machine.

Note

You can execute the install as a non-root user if you want to execute the Agent in an unprivileged user mode environment. See User Mode Installation for specific requirements and restrictions associated with an unprivileged user mode environment.

Installing Multiple Agents on a Single Machine

If you want to install multiple Agents on the same machine, you must set the following installation parameter values for each Agent being installed in addition to the initially installed Agent.

Note

You also can set these parameters values for the initial installation of an Agent on a machine.

Installation Parameter	Value
--usermode_install	**yes**
--user	A username that is different than the username specified for any other Agent installation on this machine.
--unvdir	An Agent binaries (installation) directory that is different than the installation directory specified for any other Agent installation on this machine.
--unvport	A Universal Broker port that is different than the port specified for any other Agent installation on this machine.

Changing the Default Installation Directories

By default, an Agent is installed and configured in default directories.

If you want to change these default directories, or if you want to install multiple Agents on the same machine, you must set the --usermode_install parameter to **yes** and specify new values in the following parameters. These directories must be different for each Agent on the same machine.

Note

You also can change these directories for the initial installation of an Agent on a machine.

Installation Parameter	Default Directory	Files Installed
--unvdir	**/opt/universal**	Agent binaries
--unvcfgdir	**/etc/universal**	Agent configuration files
--unvdatadir	**/var/opt/universal**	Agent data files

AIX PAM Customization

If security is set to PAM, the **pam.conf** file under **/etc** must be modified.

Below is the minimum required PAM service definition to make Universal Agent function:

ucmd	auth	required	/usr/lib/security/pam_aix
ucmd	account	required	/usr/lib/security/pam_aix

Listing Universal Agent for AIX Information

Information on installed packages is listed with the **rpm** command. The command must be executed with the superuser ID.

To list information for the Universal Agent for AIX, issue the following command:

```
rpm -qi unv
rpm -qi unv-opscli
```

Removing Universal Agent for AIX

System Install Removal

Step 1	Stop the **ubrokerd** daemon.
Step 2	Make a backup copy of the **/etc/universal** and **/var/opt/universal** directories.

Step 3	Using the superuser ID, remove all Universal Agent for AIX packages by issuing the following commands:
	`rpm -e unv-opscli`
	`rpm -e unv`
	`rm -rf /etc/universal`
	`rm -rf /var/opt/universal`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `rmuser ubroker`
Step 5	Delete the Agent group (`ubroker`): `rmgroup ubroker`

User Mode Install Removal

Step 1	Stop the Usermode Broker.
Step 2	Make a backup copy of the **<--unvcfgdir>** and **<--unvdatadir>** directories.

Step 3	Using the superuser ID, remove all Universal Agent for Linux packages by issuing the following commands:
	`rm -rf <---unvdir>`
	`rm -rf <--unvcfgdir>`
	`rm -rf <--unvdatadir>`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r [Usermode Broker user]`
Step 5	Delete the Agent group (`ubroker`): `rmgroup [Usermode Broker group]`

Agent

To remove an Agent executing in an unprivileged user mode environment (see User Mode Installation), simply stop the `ubrokerd` daemon and remove the `./universal` installation directory. To make sure that you do not mistakenly remove a system install directory, attempt the removal with a non-privileged user account or the Broker account.

Universal Agent for HP-UX Installation

- Installation Process
- Extracting Universal Agent for HP-UX Installation Files
- Distribution File
- HP-UX Package
- Installing Universal Agent for HP-UX
  - Component Selection
  - Starting the Installation Script
  - HP-UX Installation Script Parameters
  - Installation Script Example
  - User Mode Installation
  - HP-UX PAM Customization
- Listing Universal Agent for HP-UX Information
- Removing Universal Agent for HP-UX
  - System Install Removal
  - User Mode Install Removal

Installation Process

Installation of Universal Agent for HP-UX is a three-step process:

Step 1	Download product distribution file (see Universal Agent for UNIX - Distribution File).
Step 2	Extract the installation files from the distribution file.
Step 3	Install the extracted files.

Extracting Universal Agent for HP-UX Installation Files

Stonebranch, Inc. provides a product distribution file for the following HP-UX systems:

- IA64 Systems

The Universal Agent for HP-UX product distribution file is in a compressed **tar** format.

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat sb-7.2.x.x-hpux-11.23-ia64.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is **`sb-7.2.x.x-hpux-11.23-ia64.tar.Z`**.
- File is located in the current working directory.

Distribution File

The distribution file contains multiple files, including a package in the HP-UX file format (extension **.depot**). The actual base name of the **.depot** file depends on the HP-UX version for which the distribution file is intended. (See Distribution File Format for distribution file naming conventions.)

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv-7.2.x.x-hpux-11.23-ia64.depot	HP-UX .depot file-format package.
unv-opscli-7.2.x.x-hpux-11.23-ia64.depot	Universal Controller Command Line Interface (CLI) .depot file-format package.
unvinst	Installation script.

upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.

If your Universal Agent for HP-UX distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

Note

A Python 3.7 for Universal Agent distribution is not yet available for HP-UX. It will be made available in a future release.

HP-UX Package

The Universal Agent for HP-UX is packaged as a depot file (extension **.depot**). The Universal Agent depot contains product UCM.

The following table identifies the sub-products contained in the HP-UX package.

Subproduct Name	Description
UNV.UNVcom	Universal Agent common files, such as message catalogs and translation tables.
UNV.UNVctl	Universal Control Manager
UNV.UNVcts	Universal Control Server
UNV.UNVqry	Universal Query
UNV.UNVspl	Universal Spool
UNV.UNVuag	Universal Automation Center Agent
UNV.UNVubr	Universal Broker
UNV.UNVucm	Universal Command Manager
UNV.UNVucr	Universal Certificate
UNV.UNVucs	Universal Command Server
UNV.UNVudm	Universal Data Mover Manager
UNV.UNVuds	Universal Data Mover Server
UNV.UNVuem	Universal Event Monitor Manager
UNV.UNVues	Universal Event Monitor Server
UNV.UNVusp	Universal Connector for SAP
UNV.UNVutl	Universal Utilities
UNV.UNVoms	Universal Message Service (OMS)
UNV.UNVopscli	Universal Controller Command Line Interface (CLI)

Installing Universal Agent for HP-UX

Universal Agent for HP-UX is installed with the **unvinst** script, which executes the **swinstall** command. The command to start the script must be executed with the superuser ID.

Note

Stonebranch, Inc. strongly recommends the use of the **unvinst** script for the HP-UX installation above any other method.

Component Selection

The Universal Agent package contains many components, which are grouped into four categories. Components in some categories are installed and activated optionally, as specified in the **`unvinst`** script by using **`unvinst`** command line parameters.

The following table describes each category, provides the default installation configuration for the components in that category, and identifies the command line parameters to use for components that are optionally installed and activated.

Note

The default installation configuration refers to new installs only. For upgrades, installed component configurations are not changed by the upgrade process.

Category	Default Configuration	Description
Base components	Always installed.	Base components are always installed and activated. They include the Universal Broker, Universal Command (UCM), Universal Data Mover (UDM), and others. Base components provide the core agent infrastructure and workload services.
Universal Automation Center Agent (UAG)	Always installed, but inactive.	Universal Controller utilizes UAG agents to provided distributed, workload automation services. UAG is activated using the --oms_servers parameter.
Universal Message Service (OMS)	Always installed, but inactive.	OMS is message-oriented middleware that should be deployed on a small set of centrally located servers. It should not be activated on every Agent install. OMS is the network provider for UAG and the Universal Controller. OMS is activated using the --oms_autostart parameter.
Universal Controller command line programs	Not installed.	Universal Controller command line programs provide a command line interface to the Universal Controller. The installation of Universal Controller command line programs is optional. The command line programs are only required on Agents that need to interface with the Universal Controller via the command line. Universal Controller command line programs are installed using the --opscli parameter.

Starting the Installation Script

To start the installation script, **unvinst**, and install all of the HP-UX packages, issue the following command:

```
sh ./unvinst [--user username [--userdir directory] [--create_user {yes|no}] ] [--group group name [--
create_group {yes|no}] ]
             [--keystore {yes|no}] [--convert_opsagent [--opsdir directory] ] [--oms_servers network address]
[--oms_port port]
             [--ac_agent_clusters clusters] [--ac_agent_ip IP address] [--ac_netname ID]
             [--ac_process_cancel_timeout value {s|m|h|d}] [--oms_autostart {yes|no}]
             [--uag_autostart yes] [--opscli {yes|no}] [--usermode_install {yes|no} [--unvdir directory]
             [--unvcfgdir directory] [--unvdatadir directory] [--unvport port] ]
```

See HP-UX Installation Script Parameters and Installation Script Example, below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

Note

If you want to install multiple Agents on the same machine, or add one or more additional Agents to a machine with a previously installed Agent, some of these installation parameters are required (see Installing Multiple Agents on a Single Machine).

This is a silent install. The output from **unvinst** is written to file **install.log** in the current directory.

System initialization files **/sbin/init.d/ubrokerd** and **/sbin/rc3.d/S850ubrokerd** are created to start the **ubrokerd** daemon when the system boots to runlevel 3.

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the installation script parameters, below.

Note

With the Solaris, HP-UX, and AIX installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain and can be reviewed and removed, as desired, by your Administrator.

HP-UX Installation Script Parameters

The following table describes the optional parameters that are available in the installation script, **unvinst**, when installing Universal Agent.

The parameters are grouped into the following categories:

Category	Description
Base	Used for base install.
OMS	Used for an active OMS configuration.
UAG	Used for an active UAG configuration.
CLI	Used for Universal Controller CLI programs install.
User Mode	Used for user mode installation.

Parameter	Description	Default
Base Parameters		
-u **user** **--user**	Normal UNIX **username** that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Universal Agent server components (for example: **ucmsrv**, **udmsrv**, and **uemsrv**) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set. • If the user account that you want to use already exists, specify that user account. • If the user account does not exist, the install creates it. • If you want to change the user account for an installed Universal Agent for HP-UX system, you must perform a re-installation and use this parameter to change the user account. • If **--user** is omitted from **unvinst**, the default is used. • If --usermode_install is **yes**, there is no default.	**ubroker**
create_user **create_user**	Specification (**yes** or **no**) for whether or not to create the user name that will own the installed files as a local user.	**yes**
userdir **userdir**	Home directory for the created user account specified by --user. • If this directory does not exist, it is created when the specified user is created. • If the user specified by --user already exists, but the home directory of that user is not the default directory (**/home/<username>**), **--userdir** must specify the path to that home directory. • If **--userdir** is omitted from **unvinst**, the default is used.	**/home/<username>**
-g **group** **group**	Normal UNIX **groupname**; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group. • If the group that you want to use already exists, specify that group. • If this group does not exist, the install creates it. • If **--group** is omitted from **unvinst**, the default is used.	**ubroker**
create_group **create_group**	Specification (**yes** or **no**) for whether or not to create the group that will own the installed files as a local group.	**yes**
keystore **keystore**	Specification (**yes** or **no**) for whether or not to create encryption keys during installation and set up the local Universal Broker as a keystore owner.	**no**
UAG Parameters		
-c **convert** **convert_opsagent**	Causes **unvinst** to execute **opsmerge.sh** (residing in **/opt/universal/uagsrv/bin**), which performs the following tasks: 1. Searches for an existing Opswise Automation Center Agent 1.6 or 1.7 install and converts properties stored in the **agent.props** file to corresponding configuration options in the Universal Automation Center Agent (UAG) configuration file, **uags.conf**. 2. Searches for an active Opswise Automation Center Agent 1.6 or 1.7 daemon process and attempts to stop it. 3. Assigns the ID used by the Opswise Automation Center Agent 1.6 or 1.7 to UAG by moving the **qname** file from the Opswise Automaton Center install directory to the **/var/opt/universal/uag/var** directory.	

-d **opsdir** **opsdir**	If --convert_opsagent is specified: Identifies the primary install directory for Universal Agent. • If **--opsdir** is omitted from **`unvinst`**, the default is used.	**`/home`** **`/opswise`**
oms_servers **oms_servers**	Specifies a value, in the format **`port@host[,port2@host,...,portn@hostn]`**, for the port and network address of the OMS server(s) to be used as network communications providers. Note You should always include **--oms_servers** in **`unvinst`**; OMS is the network communications provider between Controller 7.2.x.x and Agent 7.2.x.x. **UAG configuration:** The value specified for **--oms_servers** is set automatically for the UAG OMS_SERVERS configuration option.	(none)
ac_agent_clusters **ac_agent_clusters**	Specifies the Universal Controller-defined clusters to which this agent will belong. **UAG configuration:** The value specified by **--ac_agent_clusters** is set automatically for the UAG AGENT_CLUSTERS configuration option.	(none)
ac_agent_ip **ac_agent_ip**	Specifies the IP address or host name (which resolves to an IP address) that the Agent reports to the Controller. **UAG configuration:** The value specified by **--ac_agent_ip** is set automatically for the UAG AGENT_IP configuration option.	(none)
ac_netname **ac_netname**	Specifies the network ID that Universal Agent will use. **UAG configuration:** The value specified by **--ac_netname** is set automatically for the UAG NETNAME configuration option.	(none)
ac_enable_ssl **ac_enable_ssl**	*This option is deprecated starting with Universal Agent 7.1.0.0. UAG Server will always attempt to use SSL/TLS for OMS connections.*	n/a
-ac_process_cancel_timeout **--ac_process_cancel_timeout**	Set the PROCESS_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time an OS task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**
uag_autostart **uag_autostart**	Specifies (**yes** or **no**) whether or not the Universal Automation Center Agent (UAG) Server starts automatically when the Universal Broker is started. **UAG component definition:** The value specified by **--uag_autostart** is set automatically for the UAG AUTOMATICALLY_START component definition option.	**yes**
OMS Parameters		
oms_port **oms_port**	Specifies the port to use to listen for OMS connection requests. **OMS configuration:** The value specified by **--oms_port** is set automatically for the OMS SERVICE_PORT configuration option.	(none)
oms_autostart **oms_autostart**	Specifies (**yes** or **no**) whether or not OMS is started automatically by Universal Broker when Universal Broker starts. **OMS component definition:** The value specified by **--oms_autostart** is set automatically for the OMS AUTOMATICALLY_START and RESTART component definition options.	**no**
CLI Parameters		

opscli **opscli**	Specifies (**yes** or **no**) whether or not the Universal Controller Command Line Interface (CLI) tools will be installed.	**no**
User Mode Parameters		
-U **-usermode_install** **--usermode_install**	Specifies (**yes** or **no**) for a user mode installation, which defines both of the following: • Whether or not this Agent can be installed on a machine where an Agent already is installed (see Installing Multiple Agents on a Single Machine). • Whether or not the user can specify installation parameters (--unvdir, --unvcfgdir, --unvdatadir, and --unvport) that control where the Agent will be installed and configured (see Changing the Default Installation Directories).	**no**
-unvdir **--unvdir**	If --usermode_install is set to **yes**: Specifies the Agent binaries (installation) directory.	(none)
unvcfgdir **unvcfgdir**	If --usermode_install is set to **yes**: Specifies the Agent configuration files directory.	**<--unvdir>/etc**
unvdatadir **unvdatadir**	If --usermode_install is set to **yes**: Specifies the Agent data files directory.	**<--unvdir>/var**
unvport **unvport**	If --usermode_install is set to **yes**: Specifies the Universal Broker port.	(none)
Additional Parameter		
-? **-h** **help** **info**	Displays command line help.	n/a

Installation Script Example

The following example illustrates Universal Agent for HP_UX installed with the installation script, **unvinst**, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup
             --keystore no --convert_opsagent --opsdir /homedir/ops
             --oms_servers 7878@oms2 --oms_port 7878 --oms_autostart yes
             --ac_agent_clusters GA Cluster,CA Cluster --ac_agent_ip 127.0.0.1
             --ac_netname OPSAUTOCONF --opscli yes
             --uag_autostart yes --usermode_install yes --unvdir /opt/universal
             --unvcfgdir /etc/universal --unvdatadir /var/opt/universal --unvport 7887
```

User Mode Installation

A user mode installation, implemented through use of the usermode_install installation parameter, lets you install multiple Agents on a single machine and change the default installation directories for any Agent being installed.

You must perform a user mode installation for installing an Agent on a machine where one or more Agents already have been installed.

However, you also can perform a user mode installation for the initial installation of an Agent on a machine.

Note

You can execute the install as a non-root user if you want to execute the Agent in an unprivileged user mode environment. See User Mode Installation for specific requirements and restrictions associated with an unprivileged user mode environment.

Installing Multiple Agents on a Single Machine

If you want to install multiple Agents on the same machine, you must set the following installation parameter values for each Agent being installed in addition to the initially installed Agent.

Note

You also can set these parameters values for the initial installation of an Agent on a machine.

Installation Parameter	Value
--usermode_install	**yes**
--user	A username that is different than the username specified for any other Agent installation on this machine.
--unvdir	An Agent binaries (installation) directory that is different than the installation directory specified for any other Agent installation on this machine.
--unvport	A Universal Broker port that is different than the port specified for any other Agent installation on this machine.

Changing the Default Installation Directories

By default, an Agent is installed and configured in default directories.

If you want to change these default directories, or if you want to install multiple Agents on the same machine, you must set the --usermode_install parameter to **yes** and specify new values in the following parameters. These directories must be different for each Agent on the same machine.

Note

You also can change these directories for the initial installation of an Agent on a machine.

Installation Parameter	Default Directory	Files Installed
--unvdir	**/opt/universal**	Agent binaries
--unvcfgdir	**/etc/universal**	Agent configuration files
--unvdatadir	**/var/opt/universal**	Agent data files

HP-UX PAM Customization

If security is set to PAM, the **pam.conf** file under **/etc** does not require modification. The **ucmd** service will use the modules defined in the "other" section of **pam.conf**.

Listing Universal Agent for HP-UX Information

On HP-UX, information on an installed product and sub-products is listed with the **swlist** command. The command must be executed with the superuser ID.

To list information for Universal Agent for HP-UX, issue the following command:

```
swlist -l subproduct OPSCLI
swlist -l subproduct UNV
```

Removing Universal Agent for HP-UX

System Install Removal

Step 1	Stop the **ubrokerd** daemon.
Step 2	Make a backup copy of the **`/etc/universal`** and **`/var/opt/universal`** directories.

Step 3	Using the superuser ID, remove all Universal Agent for HP-UX sub-products by issuing the following commands:
	`swremove OPSCLI`
	`swremove UNV`
	System initialization files **`/sbin/init.d/ubrokerd`** and **`/sbin/rc3.d/S850ubrokerd`** are removed.
	`rm -rf /etc/universal`
	`rm -rf /var/opt/universal`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r ubroker`
Step 5	Delete the Agent group (`ubroker`): `groupdel ubroker`

User Mode Install Removal

Step 1	Stop the Usermode Broker.
Step 2	Make a backup copy of the **`<--unvcfgdir>`** and **`<--unvdatadir>`** directories.

Step 3	Using the superuser ID, remove all Universal Agent for Linux packages by issuing the following commands:
	`rm -rf <---unvdir>`
	`rm -rf <--unvcfgdir>`
	`rm -rf <--unvdatadir>`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r [Usermode Broker user]`
Step 5	Delete the Agent group (`ubroker`): `groupdel [Usermode Broker group]`

Agent

To remove an Agent executing in an unprivileged user mode environment (see User Mode Installation), simply stop the `ubrokerd` daemon and remove the `./universal` installation directory. To make sure that you do not mistakenly remove a system install directory, attempt the removal with a non-privileged user account or the Broker account.

Universal Agent for Solaris Installation

- Installation Process
- Extracting the Universal Agent for Solaris Installation Files
  - Intel-Based Systems
  - SPARC-Based Systems
- Solaris Package
- Installing Universal Agent for Solaris
  - Component Selection
  - Starting the Installation Script
  - Solaris Installation Script Parameters
  - Installation Script Example
  - User Mode Installation
  - Solaris PAM Customization
- Listing Universal Agent for Solaris Information
- Removing Universal Agent for Solaris
  - System Install Removal
  - User Mode Install Removal

Installation Process

Installation of Universal Agent for Solaris is a three-step process:

Step 1	Download product distribution file (see Universal Agent for UNIX - Distribution File).
Step 2	Extract the installation files from the distribution file.
Step 3	Install the extracted files.

Extracting the Universal Agent for Solaris Installation Files

Stonebranch, Inc. provides a separate product distribution file for the following Solaris systems:

- Intel-based systems
- SPARC-based systems

Universal Agent for Solaris product distribution files are in a compressed **tar** format.

Intel-Based Systems

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat sb-7.2.x.x-solaris-10-x64.tar.Z | tar xvf -
```

This command assumes that:

- Name of the distribution file is **`sb-7.2.x.x-solaris-10-x64.tar.Z`**.
- File is located in the current working directory.

Distribution File

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv-7.2.x.x-solaris-10-x64.pkg	Solaris Intel-based file format package.
unv-opscli-7.2.x.x-solaris-10-x64.pkg	Universal Controller Command Line Interface (CLI) Intel-based file format package.
unvinst	Installation script.

upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for Solaris (Intel-based systems) distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

SPARC-Based Systems

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat sb-7.2.x.x-solaris-10-sparcv9.tar.Z | tar xvf -
```

This command assumes that:

- Name of the distribution file is **`sb-7.2.x.x-solaris-10-sparcv9.tar.Z`**.
- File is located in the current working directory.

The actual base name of the **.pkg** file depends on the Solaris version for which the distribution file is intended. (See Distribution File Format for distribution file naming conventions.)

Distribution File

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv-7.2.x.x-solaris-10-sparcv9.pkg	Solaris SPARC-based file format package.
unv-opscli-7.2.x.x-solaris-10-sparcv9.pkg	Universal Controller Command Line Interface (CLI) SPARC-based file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for Solaris (SPARC-based systems) distribution file does not contain all of these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

Solaris Package

Universal Agent for Solaris is packaged as a set of Solaris packages (extension **.pkg**).

The following table identifies the Universal Agent for Solaris package names.

Package Name	Description
UNVcom	Universal Agent common files, such as, message catalogs and translation tables.
UNVctl	Universal Control Manager
UNVcts	Universal Control Server
UNVqry	Universal Query
UNVspl	Universal Spool

UNVuag	Universal Automation Center Agent
UNVubr	Universal Broker
UNVucm	Universal Command Manager
UNVucr	Universal Certificate
UNVucs	Universal Command Server
UNVudm	Universal Data Mover Manager
UNVuds	Universal Data Mover Server
UNVuem	Universal Event Monitor Manager
UNVues	Universal Event Monitor Server
UNVusp *	Universal Connector SAP
UNVutl	Universal Utilities
UNVoms	Universal Message Service (OMS)
UNVopscli	Universal Controller Command Line Interface (CLI)

Note

UNVusp is not provided with the Universal Agent for Solaris ***Intel-based*** file format package.

Installing Universal Agent for Solaris

Universal Agent for Solaris is installed with the **unvinst** script, which executes the **pkgadd** command. The command to start the script must be executed with the superuser ID.

Note

Stonebranch, Inc. strongly recommends the use of the **unvinst** script for the Solaris installation above any other method.

Component Selection

The Universal Agent package contains many components, which are grouped into five categories. Components in some categories are installed and activated optionally, as specified in the **`unvinst`** script by using **`unvinst`** command line parameters.

The following table describes each category, provides the default installation configuration for the components in that category, and identifies the command line parameters to use for components that are optionally installed and activated.

Note

The default installation configuration refers to new installs only. For upgrades, installed component configurations are not changed by the upgrade process.

Category	Default Configuration	Description
Base components	Always installed.	Base components are always installed and activated. They include the Universal Broker, Universal Command (UCM), Universal Data Mover (UDM), and others. Base components provide the core agent infrastructure and workload services.
Universal Automation Center Agent (UAG)	Always installed, but inactive.	Universal Controller utilizes UAG agents to provided distributed, workload automation services. UAG is activated using the --oms_servers parameter.
Universal Message Service (OMS)	Always installed, but inactive.	OMS is message-oriented middleware that should be deployed on a small set of centrally located servers. It should not be activated on every Agent install. OMS is the network provider for UAG and the Universal Controller. OMS is activated using the --oms_autostart parameter.
Universal Controller command line programs	Not installed.	Universal Controller command line programs provide a command line interface to the Universal Controller. The installation of Universal Controller command line programs is optional. The command line programs are only required on Agents that need to interface with the Universal Controller via the command line. Universal Controller command line programs are installed using the --opscli parameter.

Third-party components	Not installed.	This category includes components not developed or maintained by Stonebranch, Inc. For example, Universal Agent 7.2.0.0 provides a Python 3.7 distribution that can be installed under the Universal Agent install directory. This distribution is provided so that a reliable Python environment exists on systems that may not provide for all users.

Starting the Installation Script

To start the installation script, **unvinst**, and install all of the Solaris packages, issue the following command:

```
sh ./unvinst [--user username [--userdir directory] [--create_user {yes|no}] ] [--group group name [--
create_group {yes|no}] ]
             [--keystore {yes|no}] [--convert_opsagent [--opsdir directory] ] [--oms_servers network address]
[--oms_port port]
             [--ac_agent_clusters clusters] [--ac_agent_ip IP address] [--oms_autostart {yes|no}] [--ac_netname
ID]
             [--ac_extension_accept_list list] [--ac_extension_python_list list] [--
ac_extension_deploy_on_registration {yes|no}]
             [--ac_extension_cancel_timeout value {s|m|h|d}] [--ac_process_cancel_timeout value {s|m|h|d}]
             [--uag_autostart yes] [--opscli {yes|no}] [--usermode_install {yes|no} [--unvdir directory]
             [--unvcfgdir directory] [--unvdatadir directory] [--unvport port] [--python {yes|no}] ]
```

See Solaris Installation Script Parameters and Installation Script Example, below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

Note

If you want to install multiple Agents on the same machine, or add one or more additional Agents to a machine with a previously installed Agent, some of these installation parameters are required (see Installing Multiple Agents on a Single Machine).

This is a silent install. The output from **unvinst** is written to file **install.log** in the current directory.

System initialization files **/etc/init.d/ubrokerd** and **/etc/rc3.d/S85ubrokerd** are created to start the **ubrokerd** daemon when the system boots to runlevel 3.

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the installation script parameters, below.

Note

With the Solaris, HP, and AIX installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain and can be reviewed and removed, as desired, by your Administrator.

Solaris Installation Script Parameters

The following table describes the optional parameters that are available in the installation script, **unvinst**, when installing Universal Agent.

The parameters are grouped into the following categories:

Category	Description
Base	Used for base install.
OMS	Used for an active OMS configuration.
UAG	Used for an active UAG configuration.
CLI	Used for Universal Controller CLI programs install.
User Mode	Used for user mode installation.
Third-party	Used to manage third-party components.

Parameter	Description	Default
Base Parameters		

-u **user** **--user**	Normal UNIX **username** that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Universal Agent server components (for example: **ucmsrv**, **udmsrv**, and **uemsrv**) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set. • If the user account that you want to use already exists, specify that user account. • If the user account does not exist, the install creates it. • If you want to change the user account for an installed Universal Agent for Solaris system, you must perform a re-installation and use this parameter to change the user account. • If **--user** is omitted from **unvinst**, the default is used. • If --usermode_install is **yes**, there is no default.	**ubroker**
create_user **create_user**	Specification (**yes** or **no**) for whether or not to create the user name that will own the installed files as a local user.	**yes**
userdir **userdir**	Home directory for the created user account specified by --user. • If this directory does not exist, it is created when the specified user is created. • If the user specified by --user already exists, but the home directory of that user is not the default directory (**/export/home/<username>**), **--userdir** must specify the path to that home directory. • If **--userdir** is omitted from **unvinst**, the default is used.	**/export** **/home** **/<username>**
-g **group** **group**	Normal UNIX **groupname**; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group. • If the group that you want to use already exists, specify that group. • If this group does not exist, the install creates it. • If **--group** is omitted from **unvinst**, the default is used.	**ubroker**
create_group **create_group**	Specification (**yes** or **no**) for whether or not to create the group that will own the installed files as a local group.	**yes**
keystore **keystore**	Specification (**yes** or **no**) for whether or not to create encryption keys during installation and set up the local Universal Broker as a keystore owner.	**no**
UAG Parameters		
-c **convert** **convert_opsagent**	Causes **unvinst** to execute **opsmerge.sh** (residing in **/opt/universal/uagsrv/bin**), which performs the following tasks: 1. Searches for an existing Opswise Automation Center Agent 1.6 or 1.7 install and converts properties stored in the **agent.props** file to corresponding configuration options in the Universal Automation Center Agent (UAG) configuration file, **uags.conf**. 2. Searches for an active Opswise Automation Center Agent 1.6 or 1.7 daemon process and attempts to stop it. 3. Assigns the ID used by the Opswise Automation Center Agent 1.6 or 1.7 to UAG by moving the **qname** file from the Opswise Automaton Center install directory to the **/var/opt/universal/uag/var** directory.	
-d **opsdir** **opsdir**	If --convert_opsagent is specified: Identifies the primary install directory for Universal Agent. • If **--opsdir** is omitted from **unvinst**, the default is used.	**/export** **/home** **/opswise**
oms_servers **oms_servers**	Specifies a value, in the format **port@host[,port2@host,...,portn@hostn]**, for the port and network address of the OMS server(s) to be used as network communications providers. Note You should always include **--oms_servers** in **unvinst**; OMS is the network communications provider between Controller 7.2.x.x and Agent 7.2.x.x. **UAG configuration:** The value specified for **--oms_servers** is set automatically for the UAG OMS_SERVERS configuration option.	(none)
ac_agent_clusters **ac_agent_clusters**	Specifies the Universal Controller-defined clusters to which this agent will belong. **UAG configuration:** The value specified by **--ac_agent_clusters** is set automatically for the UAG AGENT_CLUSTERS configuration option.	(none)

ac_agent_ip **ac_agent_ip**	Specifies the IP address or host name (which resolves to an IP address) that the Agent reports to the Controller. **UAG configuration:** The value specified by **--ac_agent_ip** is set automatically for the UAG AGENT_IP configuration option.	(none)
ac_netname **ac_netname**	Specifies the network ID that Universal Agent will use. **UAG configuration:** The value specified by **--ac_netname** is set automatically for the UAG NETNAME configuration option.	(none)
-ac_extension_accept_list **--ac_extension_accept_list**	Specifies a comma-separated list of one or more Universal Extensions that the agent will accept via auto-deployment from the Controller. • A single value of ***** indicates that all extensions are accepted. • A single value of **none** indicates that no extensions are accepted. The value specified for this parameter sets the EXTENSION_ACCEPT_LIST configuration option value.	*****
-ac_extension_python_list --ac_extension_python_list	Specifies a comma-separated list of zero or more Python locations. Each item in the list is expected to contain a complete path to a Python executable. The value specified for this parameter sets the EXTENSION_PYTHON_LIST configuration option value.	/usr/bin/python3,/usr/bin/python,/usr/libexec/platform-python
-ac_extension_deploy_on_registration --ac_extension_deploy_on_registration	Controls Extension deployment behavior from the Universal Controller. • If the value is **yes**, the Controller will preemptively deploy all extensions acceptable by UAG. • If the value is **no**, the Controller will only send Extension modules as needed (on demand). The value specified for this parameter sets the EXTENSION_DEPLOY_ON_REGISTRATION configuration option value.	**no**
-ac_extension_cancel_timeout **--ac_extension_cancel_timeout**	Set the EXTENSION_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time a Universal Extension task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**
ac_enable_ssl **ac_enable_ssl**	*This option is deprecated starting with Universal Agent 7.1.0.0. UAG Server will always attempt to use SSL/TLS for OMS connections.*	n/a
-ac_process_cancel_timeout **--ac_process_cancel_timeout**	Set the PROCESS_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time an OS task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**

uag_autostart **uag_autostart**	Specifies (**yes** or **no**) whether or not the Universal Automation Center Agent (UAG) Server starts automatically when the Universal Broker is started. **UAG component definition:** The value specified by **--uag_autostart** is set automatically for the UAG AUTOMATICALLY_START component definition option.	**yes**
OMS Parameters		
oms_port **oms_port**	Specifies the port to use to listen for OMS connection requests. **OMS configuration:** The value specified by **--oms_port** is set automatically for the OMS SERVICE_PORT configuration option.	(none)
oms_autostart **oms_autostart**	Specifies (**yes** or **no**) whether or not OMS is started automatically by Universal Broker when Universal Broker starts. **OMS component definition:** The value specified by **--oms_autostart** is set automatically for the OMS AUTOMATICALLY_START and RESTART component definition options.	**no**
CLI Parameters		
opscli **opscli**	Specifies (**yes** or **no**) whether or not the Universal Controller Command Line Interface (CLI) tools will be installed.	**no**
User Mode Parameters		
-U **-usermode_install** **--usermode_install**	Specifies (**yes** or **no**) for a user mode installation, which defines both of the following: • Whether or not this Agent can be installed on a machine where an Agent already is installed (see Installing Multiple Agents on a Single Machine). • Whether or not the user can specify installation parameters (--unvdir, --unvcfgdir, --unvdatadir, and --unvport) that control where the Agent will be installed and configured (see Changing the Default Installation Directories).	**no**
-unvdir **--unvdir**	If --usermode_install is set to **yes**: Specifies the Agent binaries (installation) directory.	(none)
unvcfgdir **unvcfgdir**	If --usermode_install is set to **yes**: Specifies the Agent configuration files directory.	**<--unvdir>/etc**
unvdatadir **unvdatadir**	If --usermode_install is set to **yes**: Specifies the Agent data files directory.	**<--unvdir>/var**
unvport **unvport**	If --usermode_install is set to **yes**: Specifies the Universal Broker port.	(none)
Third-Party Parameters		
python **python**	Specifies (**yes** or **no**) whether the Python 3.7 Distribution for Universal Agent is installed. Note If **yes**, libintl.so version 0.18.1.1, REV=2011.03.15 or higher will be required. The install will proceed if the required dependency does not exist, but the python executable provided by this distribution will not run until libintl is available	**no**
Additional Parameter		
-? **-h** **help** **info**	Displays command line help.	n/a

Installation Script Example

The following example illustrates Universal Agent for Solaris installed with the installation script, **unvinst**, and optional parameters.
&nbap;

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup --keystore no --convert_opsagent
             --opsdir /homedir/ops --oms_servers 7878@oms2 --oms_port 7878
             --oms_autostart yes --ac_agent_clusters GA Cluster,CA Cluster --ac_agent_ip 127.0.0.1
             --ac_netname OPSAUTOCONF --opscli yes --uag_autostart yes --usermode_install yes
             --unvdir /opt/universal --unvcfgdir /etc/universal --unvdatadir /var/opt/universal --unvport 7887
```

User Mode Installation

A user mode installation, implemented through use of the usermode_install installation parameter, lets you install multiple Agents on a single machine and change the default installation directories for any Agent being installed.

You must perform a user mode installation for installing an Agent on a machine where one or more Agents already have been installed.

However, you also can perform a user mode installation for the initial installation of an Agent on a machine.

Note

You can execute the install as a non-root user if you want to execute the Agent in an unprivileged user mode environment. See User Mode Installation for specific requirements and restrictions associated with an unprivileged user mode environment.

Installing Multiple Agents on a Single Machine

If you want to install multiple Agents on the same machine, you must set the following installation parameter values for each Agent being installed in addition to the initially installed Agent.

Note

You also can set these parameters values for the initial installation of an Agent on a machine.

Installation Parameter	Value
--usermode_install	**yes**
--user	A username that is different than the username specified for any other Agent installation on this machine.
--unvdir	An Agent binaries (installation) directory that is different than the installation directory specified for any other Agent installation on this machine.
--unvport	A Universal Broker port that is different than the port specified for any other Agent installation on this machine.

Changing the Default Installation Directories

By default, an Agent is installed and configured in default directories.

If you want to change these default directories, or if you want to install multiple Agents on the same machine, you must set the --usermode_install parameter to **yes** and specify new values in the following parameters. These directories must be different for each Agent on the same machine.

Note

You also can change these directories for the initial installation of an Agent on a machine.

Installation Parameter	Default Directory	Files Installed
--unvdir	**/opt/universal**	Agent binaries
--unvcfgdir	**/etc/universal**	Agent configuration files
--unvdatadir	**/var/opt/universal**	Agent data files

Solaris PAM Customization

If security is set to PAM, the **pam.conf** file under **/etc** does not require modification. The **ucmd** service will use the modules defined in the "other" section of **pam.conf**.

Listing Universal Agent for Solaris Information

Information on an installed packages is listed with the **pkginfo** command. The command must be executed with the superuser ID.

To list information for Universal Agent for Solaris, issue the following command:

```
pkginfo UNVopscli UNVoms UNVucr UNVusp UNVutl UNVues UNVuem UNVuds UNVudm UNVucs UNVucm UNVqry UNVctl UNVcts
UNVuag UNVubr UNVspl UNVcom
```

Note

UNVusp is not provided with the Universal Agent for Solaris Intel-based file format package.

Removing Universal Agent for Solaris

System Install Removal

Step 1	Stop the **ubrokerd** daemon.
Step 2	Make a backup copy of the **/var/opt/universal** and **/etc/universal** directories.

Step 3	Using the superuser ID, remove all Universal Agent for Solaris packages by issuing the following commands:
	`pkgrm UNVopscli UNVoms UNVucr UNVusp UNVutl UNVues UNVuem UNVuds UNVudm UNVucs UNVucm UNVqry UNVctl UNVcts UNVuag UNVubr UNVspl UNVcom`
	System initialization files **/etc/init.d/ubrokerd** and **/etc/rc3.d/S85ubrokerd** are removed.
	Note
	UNVusp is not provided with the Universal Agent for Solaris Intel-based file format package.
	`rm -rf /etc/universal`
	`rm -rf /var/opt/universal`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r ubroker`
Step 5	Delete the Agent group (`ubroker`): `groupdel ubroker`

User Mode Install Removal

Step 1	Stop the Usermode Broker.
Step 2	Make a backup copy of the **<--unvcfgdir>** and **<--unvdatadir>** directories.

Step 3	Using the superuser ID, remove all Universal Agent for Linux packages by issuing the following commands:
	`rm -rf <---unvdir>`
	`rm -rf <--unvcfgdir>`
	`rm -rf <--unvdatadir>`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r [Usermode Broker user]`
Step 5	Delete the Agent group (`ubroker`): `groupdel [Usermode Broker group]`

Agent

To remove an Agent executing in an unprivileged user mode environment (see User Mode Installation), simply stop the `ubrokerd` daemon and remove the `./universal` installation directory. To make sure that you do not mistakenly remove a system install directory, attempt the removal with a non-privileged user account or the Broker account.

Universal Agent for Linux Installation

- Installation Process
- Extracting the Universal Agent for Linux Installation Files
  - x86_64-Based Systems
  - IBM S/390 and zSeries Systems
  - Debian-Based Systems
  - PowerPC-Based Systems
- Installing Universal Agent for Linux
  - Component Selection
  - Starting the Installation Script
  - Linux Installation Script Parameters
  - Installation Script Example
  - User Mode Installation
- Listing Universal Agent for Linux Information
  - RPM Commands
  - Debian Commands
- Removing Universal Agent for Linux
  - System Install Removal
  - User Mode Install Removal
- Linux PAM Customization
  - PAM Configuration File
  - Examples of Customized PAM Configuration Files
- Configuring the Agent to Run a Task without a Password

Installation Process

Installation of Universal Agent for Linux is a three-step process:

Step 1	Download product distribution file (see Universal Agent for UNIX - Distribution File).
Step 2	Extract the installation files from the distribution file.
Step 3	Install the extracted files.

Extracting the Universal Agent for Linux Installation Files

Stonebranch, Inc. provides separate product distribution files for the following Linux systems:

- x86_64-based systems
- IBM S/390 and zSeries systems
- Debian-based systems
- PowerPC-based systems

Universal Agent for Linux product distribution files are in a compressed **tar** format.

x86_64-Based Systems

(The RPM package for x86_64-based systems contains the **x86_64** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-7.2.x.x-linux-3.10-x86_64.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **`sb-7.2.x.x-linux-3.10-x86_64.tar.Z`**.

Distribution File

The following table identifies the files contained in the distribution file.

File	Description

Readme.unv	Summary of the installation procedure.
unv-7.2.x.x-linux-3.10-x86_64.rpm	Linux RPM file format package.
unv-opscli-7.2.x.x-linux-3.10-x86_64.rpm	Universal Controller Command Line Interface (CLI) RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for Linux (x86_64-based systems) distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

IBM S/390 and zSeries Systems

(The RPM package for IBM S/390 and zArchitecture systems contains the **s390x** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-7.2.x.x-linux-3.10-s390x.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **`sb-7.2.x.x-linux-3.10-s390x.tar.Z`**.

Distribution File

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary the installation procedure.
unv-7.2.x.x-linux-3.10-s390x.rpm	Linux RPM file format package.
unv-opscli-7.2.x.x-linux-3.10-s390x.rpm	Universal Controller Command Line Interface (CLI) RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for Linux (IBM S/390 and zSeries systems) distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

Debian-Based Systems

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-7.2.x.x-linux-3-x86_64-deb.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **`sb-7.2.x.x-linux-3-x86_64-deb.tar.Z`**.

Distribution File

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary the installation procedure.
unv-7.2.x.x-linux-3-x86_64-deb	Linux RPM file format package.
unv-opscli-7.2.x.x-linux-3-x86_64.deb	Universal Controller Command Line Interface (CLI) RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for Linux (Debian-based systems) distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

PowerPC-Based Systems

(The RPM package for PowerPC-Based systems contains the **ppc64** qualifier. A further qualifier, **le**, identifies it as Little Endian).

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-7.2.x.x-linux-3.10-ppc64le.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **`sb-7.2.x.x-linux-3.10-ppc64le.tar.Z`**.

Distribution File

The following table identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary the installation procedure.
unv-7.2.x.x-linux-3.10-ppc64le.rpm	Linux RPM file format package.
unv-opscli-7.2.x.x-linux-3.10-ppc64le.rpm	Universal Controller Command Line Interface (CLI) RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.
usrmode.inc	User-mode installer main script.
unvfiles.tar	User-mode installer modules archive; a set of scripts loaded and executed by **usrmode.inc**.
unv-python3.7.tar	Python 3.7 Distribution for Universal Agent.

Note

If your Universal Agent for Linux (PowerPC-based systems) distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

Installing Universal Agent for Linux

Universal Agent for Linux is installed with the **unvinst** script, which executes the **rpm** command. The command to start the script must be executed with the superuser ID.

Note

Stonebranch, Inc. strongly recommends the use of the **unvinst** script for the Linux installation above any other method.

Component Selection

The Universal Agent package contains many components, which are grouped into five categories. Components in some categories are installed and activated optionally, as specified in the **unvinst** script by using **unvinst** command line parameters.

The following table describes each category, provides the default installation configuration for the components in that category, and identifies the command line parameters to use for components that are optionally installed and activated.

Note

The default installation configuration refers to new installs only. For upgrades, installed component configurations are not changed by the upgrade process.

Category	Default Configuration	Description
Base components	Always installed.	Base components are always installed and activated. They include the Universal Broker, Universal Command (UCM), Universal Data Mover (UDM), and others. Base components provide the core agent infrastructure and workload services.
Universal Automation Center Agent (UAG)	Always installed, but inactive.	Universal Controller utilizes UAG agents to provided distributed, workload automation services. UAG is activated using the --oms_servers parameter.
Universal Message Service (OMS)	Always installed, but inactive.	OMS is message-oriented middleware that should be deployed on a small set of centrally located servers. It should not be activated on every Agent install. OMS is the network provider for UAG and the Universal Controller. OMS is activated using the --oms_autostart parameter.
Universal Controller command line programs	Not installed.	Universal Controller command line programs provide a command line interface to the Universal Controller. The installation of Universal Controller command line programs is optional. The command line programs are only required on Agents that need to interface with the Universal Controller via the command line. Universal Controller command line programs are installed using the --opscli parameter.
Third-party components	Not installed.	This category includes components not developed or maintained by Stonebranch, Inc. For example, Universal Agent 7.2.0.0 provides a Python 3.7 distribution that can be installed under the Universal Agent install directory. This distribution is provided so that a reliable Python environment exists on systems that may not provide for all users.

Starting the Installation Script

To start the installation script, **unvinst**, issue the following command:

```
sh ./unvinst [--user username [--userdir directory] [--create_user {yes|no}] ] [--group group name [--
create_group {yes|no}] ]
             [--keystore {yes|no}] [--security {appdef|default|inherit|pam|pam_sessions}]
             [--convert_opsagent [--opsdir directory] ] [--oms_servers network address] [--oms_port port] [--
ac_agent_clusters clusters]
             [--ac_agent_ip IP address] [--oms_autostart {yes|no}] [--ac_netname ID] [--uag_autostart yes]
             [--ac_extension_accept_list list] [--ac_extension_python_list list] [--
ac_extension_deploy_on_registration {yes|no}]
             [--ac_extension_cancel_timeout value {s|m|h|d}] [--ac_process_cancel_timeout value {s|m|h|d}]
             [--opscli {yes|no}] [--usermode_install {yes|no} [--unvdir directory] [--unvcfgdir directory]
             [--unvdatadir directory] [--unvport port] [--python {yes|no}] ]
```

See Linux Installation Script Parameters and Installation Script Example, below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

Note

If you want to install multiple Agents on the same machine, or add one or more additional Agents to a machine with a previously installed Agent, some of these installation parameters are required (see Installing Multiple Agents on a Single Machine).

This is a silent install. The output from **unvinst** is written to file **install.log** in the current directory. The Agent is installed into directory **/opt**.

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the installation script parameters, below.

Note

For this release of Linux RPM, the previous non-RPM version will not be uninstalled. The Administrator can remove the remaining files/directories as needed.

Linux Installation Script Parameters

The following table describes the optional parameters that are available in the UNIX install script (**`unvinst`**) when installing Universal Agent.

The parameters are grouped into the following categories:

Category	Description
Base	Used for base install.
OMS	Used for an active OMS configuration.
UAG	Used for an active UAG configuration.
CLI	Used for Universal Controller CLI programs install.
User Mode	Used for user mode installation.
Third-party	Used to manage third-party components.

Parameter	Description	Default
Base Parameters		
-u **user** **--user**	Normal UNIX **username** that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Universal Agent server components (for example: **ucmsrv**, **udmsrv**, and **uemsrv**) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set. • If the user account that you want to use already exists, specify that user account. • If the user account does not exist, the install creates it. • If you want to change the user account for an installed Universal Agent for Linux system, you must perform a re-installation and use this parameter to change the user account. • If **--user** is omitted from **`unvinst`**, the default is used. • If --usermode_install is **yes**, there is no default.	**ubroker**
create_user **create_user**	Specification (**yes** or **no**) for whether or not to create the user name that will own the installed files as a local user.	**yes**
userdir **userdir**	Home directory for the created user account specified by --user. • If this directory does not exist, it is created when the specified user is created. • If the user specified by --user already exists, but the home directory of that user is not the default directory (**`/home/<username>`**), **`--userdir`** must specify the path to that home directory. • If **--userdir** is omitted from **`unvinst`**, the default is used.	**`/home`** **`/<username>`**
-g **group** **group**	Normal UNIX **groupname**; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group. • If the group that you want to use already exists, specify that group. • If this group does not exist, the install creates it. • If **--group** is omitted from **`unvinst`**, the default is used.	**ubroker**
create_group **create_group**	Specification (**yes** or **no**) for whether or not to create the group that will own the installed files as a local group.	**yes**
keystore **keystore**	Specification (**yes** or **no**) for whether or not to create encryption keys during installation and set up the local Universal Broker as a keystore owner.	**no**

-security **--security**	Sets the value (default, inherit, pam, pam_sessions, or trusted) of the following configuration options for Universal Agent server components: • SECURITY: UAG Server • USER_SECURITY: UCMD Server • USER_SECURITY: UCTL Server • USER_SECURITY: UDM Server • USER_SECURITY: UEM Server The appdef value for -security controls how product configuration options are set; it does not map to a product configuration option. Note pam_sessions is a valid value only for the UCMD Server and UDM Server.	(none)
-ubroker_start **--ubroker_start**	Specifies (yes or no) whether or not Universal Broker is started by the Installer when installation is completed. If --ubroker_start is not included in the script, Universal Broker will be started when installation is complete. Note --ubroker_start is ignored for user mode installs.	**yes**
UAG Parameters		
-c **convert** **convert_opsagent**	Causes **`unvinst`** to execute **opsmerge.sh** (residing in **`/opt/universal/uagsrv/bin`**), which performs the following tasks: 1. Searches for an existing Opswise Automation Center Agent 1.6 or 1.7 install and converts properties stored in the **agent.props** file to corresponding configuration options in the Universal Automation Center Agent (UAG) configuration file, **`uags.conf`**. 2. Searches for an active Opswise Automation Center Agent 1.6 or 1.7 daemon process and attempts to stop it. 3. Assigns the ID used by the Opswise Automation Center Agent 1.6 or 1.7 to UAG by moving the **qname** file from the Opswise Automaton Center install directory to the **`/var/opt/universal/uag/var`** directory.	
-d **opsdir** **opsdir**	If --convert_opsagent is specified: Identifies the primary install directory for Universal Agent. • If **--opsdir** is omitted from **`unvinst`**, the default is used.	**`/home/opswise`**
oms_servers **oms_servers**	Specifies a value, in the format **`port@host[,port2@host,...,portn@hostn]`**, for the port and network address of the OMS server(s) to be used as network communications providers. Note You should always include **--oms_servers** in **`unvinst`**; OMS is the network communications provider between a Controller 7.2.x.x and Agent 7.2.x.x. **UAG configuration:** The value specified for **--oms_servers** is set automatically for the UAG OMS_SERVERS configuration option.	(none)
ac_agent_clusters **ac_agent_clusters**	Specifies the Universal Controller-defined clusters to which this agent will belong. **UAG configuration:** The value specified by **--ac_agent_clusters** is set automatically for the UAG AGENT_CLUSTERS configuration option.	(none)
ac_agent_ip **ac_agent_ip**	Specifies the IP address or host name (which resolves to an IP address) that the Agent reports to the Controller. **UAG configuration:** The value specified by **--ac_agent_ip** is set automatically for the UAG AGENT_IP configuration option.	(none)
ac_netname **ac_netname**	Specifies the network ID that Universal Agent will use. **UAG configuration:** The value specified by **--ac_netname** is set automatically for the UAG NETNAME configuration option.	(none)

-ac_extension_accept_list --ac_extension_accept_list	Specifies a comma-separated list of one or more Universal Extensions that the agent will accept via auto-deployment from the Controller. • A single value of ***** indicates that all extensions are accepted. • A single value of **none** indicates that no extensions are accepted. The value specified for this parameter sets the EXTENSION_ACCEPT_LIST configuration option value.	*****
-ac_extension_python_list --ac_extension_python_list	Specifies a comma-separated list of zero or more Python locations. Each item in the list is expected to contain a complete path to a Python executable. The value specified for this parameter sets the EXTENSION_PYTHON_LIST configuration option value.	/usr/bin/python3,/usr/bin/python,/usr/libexec/platform-python
-ac_extension_deploy_on_registration --ac_extension_deploy_on_registration	Controls Extension deployment behavior from the Universal Controller. • If the value is **yes**, the Controller will preemptively deploy all extensions acceptable by UAG. • If the value is **no**, the Controller will only send Extension modules as needed (on demand). The value specified for this parameter sets the EXTENSION_DEPLOY_ON_REGISTRATION configuration option value.	**no**
-ac_extension_cancel_timeout --ac_extension_cancel_timeout	Set the EXTENSION_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time a Universal Extension task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**
-ac_process_cancel_timeout --ac_process_cancel_timeout	Set the PROCESS_CANCEL_TIMEOUT UAG configuration option, which specifies the length of time an OS task is given to complete its response to a CANCEL request. If the task fails to finish its own termination processing within the specified timeout period, UAG Server will forcefully terminate the task. The specified timeout must be numeric, but a one-letter suffix is accepted to specify (**s**)econds, (**m**)inutes, (**h**)ours, or (**d**)ays. If no time unit is specified, the default is seconds. The following maximums are enforced: • 2147483647 or 2147483647s • 35791394m • 596523h • 24855d Minute, hour, and day maximums are set to ensure that their value represented as a number of seconds does not exceed 2147483647.	**10**
ac_enable_ssl ac_enable_ssl	*This option is deprecated starting with Universal Agent 7.1.0.0. UAG Server will always attempt to use SSL/TLS for OMS connections.*	n/a
uag_autostart uag_autostart	Specifies (**yes** or **no**) whether or not the Universal Automation Center Agent (UAG) Server starts automatically when the Universal Broker is started. **UAG component definition:** The value specified by **--uag_autostart** is set automatically for the UAG AUTOMATICALLY_START component definition option.	**yes**
OMS Parameters		
oms_port oms_port	Specifies the port to use to listen for OMS connection requests. **OMS configuration:** The value specified by **--oms_port** is set automatically for the OMS SERVICE_PORT configuration option.	(none)

oms_autostart **oms_autostart**	Specifies (**yes** or **no**) whether or not OMS is started automatically by Universal Broker when Universal Broker starts. **OMS component definition:** The value specified by **--oms_autostart** is set automatically for the OMS AUTOMATICALLY_START and RESTART component definition options.	**no**
CLI Parameters		
opscli **opscli**	Specifies (**yes** or **no**) whether or not the Universal Controller Command Line Interface (CLI) tools will be installed.	**no**
User Mode Parameters		
-U **-usermode_install** **--usermode_install**	Specifies (**yes** or **no**) for a user mode installation, which defines both of the following: • Whether or not this Agent can be installed on a machine where an Agent already is installed (see Installing Multiple Agents on a Single Machine). • Whether or not the user can specify installation parameters (--unvdir, --unvcfgdir, --unvdatadir, and --unvport) that control where the Agent will be installed and configured (see Changing the Default Installation Directories).	**no**
-unvdir **--unvdir**	If --usermode_install is set to **yes**: Specifies the Agent binaries (installation) directory.	(none)
unvcfgdir **unvcfgdir**	If --usermode_install is set to **yes**: Specifies the Agent configuration files directory.	**<--unvdir>/etc**
unvdatadir **unvdatadir**	If --usermode_install is set to **yes**: Specifies the Agent data files directory.	**<--unvdir>/var**
unvport **unvport**	If --usermode_install is set to **yes**: Specifies the Universal Broker port.	(none)
Third-Party Parameters		
python **python**	Specifies (**yes** or **no**) whether the Python 3.7 Distribution for Universal Agent is installed.	**no**
Additional Parameter		
-? **-h** **help** **info**	Displays command line help.	n/a

Installation Script Example

The following example illustrates Universal Agent for Linux installed with the installation script, **unvinst**, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup
             --keystore no --convert_opsagent --opsdir /homedir/ops
             --oms_servers 7878@oms2 --oms_port 7878 --oms_autostart yes
             --ac_agent_clusters GA Cluster,CA Cluster --ac_agent_ip 127.0.0.1
             --ac_netname OPSAUTOCONF --opscli yes --uag_autostart yes
             --usermode_install yes --unvdir /opt/universal
             --unvcfgdir /etc/universal --unvdatadir /var/opt/universal --unvport 7887
```

User Mode Installation

A user mode installation, implemented through use of the usermode_install installation parameter, lets you install multiple Agents on a single machine and change the default installation directories for any Agent being installed.

You must perform a user mode installation for installing an Agent on a machine where one or more Agents already have been installed.

However, you also can perform a user mode installation for the initial installation of an Agent on a machine.

Note

You can execute the install as a non-root user if you want to execute the Agent in an unprivileged user mode environment. See User Mode Installation for specific requirements and restrictions associated with an unprivileged user mode environment.

Installing Multiple Agents on a Single Machine

If you want to install multiple Agents on the same machine, you must set the following installation parameter values for each Agent being installed in addition to the initially installed Agent.

Note

You also can set these parameters values for the initial installation of an Agent on a machine.

Installation Parameter	Value
--usermode_install	**yes**
--user	A username that is different than the username specified for any other Agent installation on this machine.
--unvdir	An Agent binaries (installation) directory that is different than the installation directory specified for any other Agent installation on this machine.
--unvport	A Universal Broker port that is different than the port specified for any other Agent installation on this machine.

Changing the Default Installation Directories

By default, an Agent is installed and configured in default directories.

If you want to change these default directories, or if you want to install multiple Agents on the same machine, you must set the --usermode_install parameter to **yes** and specify new values in the following parameters. These directories must be different for each Agent on the same machine.

Note

You also can change these directories for the initial installation of an Agent on a machine.

Installation Parameter	Default Directory	Files Installed
--unvdir	**/opt/universal**	Agent binaries
--unvcfgdir	**/etc/universal**	Agent configuration files
--unvdatadir	**/var/opt/universal**	Agent data files

Listing Universal Agent for Linux Information

RPM Commands

Information on installed packages is listed with the **rpm** command. The command must be executed with the superuser ID.

To list information for the Universal Agent for Linux, issue the following command:

```
rpm -qi unv
rpm -qi unv-opscli
```

Debian Commands

Information on installed packages is listed with the **dpkg** command. The command must be executed with superuser authority.

To list information for the Universal Agent for Linux, issue the following commands:

```
sudo dpkg -l unv
sudo dpkg -l unv-opscli
```

Removing Universal Agent for Linux

System Install Removal

Step 1	Stop the **ubrokerd** daemon.
Step 2	Make a backup copy of the **`/var/opt/universal`** and **`/etc/universal`** directories.

Step 3	**RPM Commands**
	
	Using the superuser ID, remove all Universal Agent for Linux packages by issuing the following commands:
	
	`rpm -e unv-opscli`
	`rpm -e unv`
	
	`rm -rf /etc/universal`
	
	`rm -rf /var/opt/universal`
	
	Debian commands
	
	Using the superuser authority, remove all Universal Agent for Linux packages by issuing the following commands:
	
	`sudo dpkg -r unv-opscli`
	`sudo dpkg -r unv`
	
	`rm -rf /etc/universal`
	
	`rm -rf /var/opt/universal`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r ubroker`

Step 5	Delete the Agent group (`ubroker`): `groupdel ubroker`

User Mode Install Removal

Step 1	Stop the Usermode Broker.
Step 2	Make a backup copy of the **<--unvcfgdir>** and **<--unvdatadir>** directories.

Step 3	Using the superuser ID, remove all Universal Agent for Linux packages by issuing the following commands:
	`rm -rf <---unvdir>`
	`rm -rf <--unvcfgdir>`
	`rm -rf <--unvdatadir>`

Step 4	Delete the Agent user account (`ubroker`) and its home directory: `userdel -r [Usermode Broker user]`
Step 5	Delete the Agent group (`ubroker`): `groupdel [Usermode Broker group]`

Agent

To remove an Agent executing in an unprivileged user mode environment (see User Mode Installation), simply stop the `ubrokerd` daemon and remove the `./universal` installation directory. To make sure that you do not mistakenly remove a system install directory, attempt the removal with a non-privileged user account or the Broker account.

Linux PAM Customization

Linux installations utilize Pluggable Authentication Modules (PAM) for user authentication. Many of the Universal Agent servers, such as Universal Command (UCMD), Universal Data Mover (UDM), and Universal Control (UCTL), must authenticate user accounts and passwords. Proper PAM configuration is essential for product operation.

There are many organizations and companies that package and distribute the Linux operating system. Most have consistent PAM implementations, but there are exceptions.

All supported Linux installations - except for SUSE 9 and below - require the sample PAM configuration file delivered with Universal Agent to be copied to directory **/etc/pam.d**:

```
cp /opt/universal/ucmdsrv/samp/ucmd.pam /etc/pam.d/ucmd
```

PAM Configuration File

All Universal Agent components utilize the same PAM configuration file.

64-Bit Linux Systems

For 64-bit Linux systems (x86_64-Based systems; S/390 and z/Series systems) its contents are:

auth	required	/lib64/security/pam_pwdb.so shadow nullok
auth	required	/lib64/security/pam_nologin.so
account	required	/lib64/security/pam_pwdb.so

Your Administrator must modify this sample PAM file to meet your local configuration.

Examples of Customized PAM Configuration Files

Universal Agent for Redhat systems 5.0 and greater

auth	include	system-auth
auth	required	pam_nologin.so
account	include	system-auth

Universal Agent for SUSE-based systems 10.0 and greater

auth	required	pam_unix2.so nullok
auth	required	pam_nologin.so
account	include	common-account

Alternative Universal Agent for SUSE 10.1

auth	required	pam_unix2.so nullok
auth	required	pam_nologin.so
account	include	common-account

Configuring the Agent to Run a Task without a Password

Prior to release 5.1, Universal Automation Center used the **ops_suexec.nopass file**, which listed all trusted users.

As of release 5.1, this file no longer is used. To apply your desired security configuration, access the UAG SECURITY configuration option in the **`uags.conf`** configuration file:

- If you do not want security, set the value to **none**.
- If you want security, set the value to **pam** and update the following files:
  - Check the **/etc/pam.d/ucmd** configuration file to ensure that it contains the minimum PAM security settings (see Examples of Customized PAM Configuration Files, above).
  - Add the following to **/etc/universal/uacl.conf** for each user: **uag_work_request [username],allow,noauth**
    Also, verify that the user has a **/home** directory defined.

Universal Agent for UNIX - Customization

- Overview
- Universal Broker Customization
  - Universal Broker Configuration
  - Universal Broker System Initialization
- Universal Automation Center Agent Customization
  - Universal Automation Center Agent Configuration
- Universal Command Manager Customization
  - Universal Command Manager Configuration
- Universal Command Server Customization
  - Universal Command Server Configuration
- Universal Connector Customization
  - Universal Connector Configuration
  - Universal Connector SAP RFC Configuration
- Universal Control Manager Customization
  - Universal Control Manager Configuration
- Universal Control Server Customization
  - Universal Control Server Configuration
- Universal Data Mover Manager Customization
  - Universal Data Mover Manager Configuration
- Universal Data Mover Server Customization
  - Universal Data Mover Server Configuration
- Universal Event Monitor Manager Customization
  - Universal Event Monitor Manager Configuration
- Universal Event Monitor Server Customization
  - Universal Event Monitor Server Configuration
- Universal Query Customization
  - Universal Query Configuration

Overview

This page provides the following information for the customization of Universal Agent components:

- Configuration
- System initialization (Universal Broker)

(For information on applying product licenses to installed Universal Agent for UNIX components, see UNIX Installation - Licensing.)

Universal Broker Customization

Universal Broker Configuration

Configuration options for Universal Broker are stored in configuration file, **ubroker.conf**, in directory **/etc/universal** by default.

See the Universal Broker 7.2.x Reference Guide for details on configuring Universal Broker.

Universal Broker System Initialization

A Broker daemon start-up script is provided in file **`/opt/universal/ubroker/ubrokerd`**. A single command line argument — either **start**, **stop**, **status**, or **restart** — instructs the script on the action to take.

See the Universal Broker 7.2.x Reference Guide for details on the Broker daemon script.

To start the Broker daemon automatically each time that the system is started, add this script to the system initialization process. This is performed by the AIX, HP-UX, Solaris, and Linux (Red Hat- and SUSE-style systems) installation process, but not the generic UNIX installation script.

Unless the Broker daemon is started by the system initialization process, it will inherit the environment of the user that starts it. In some cases, there may be environmental variables that should not be inherited. For those variables, the start-up script **`/opt/universal/ubroker-7.2.x/ubrokerd`** can be modified to unset the undesired environmental variables.

The format of the unset is as follows:

- **unset Variable1**
- **unset MAIL**
- **unset Variable2**

This above modifications would cause **Variable1**, **MAIL**, and **Variable2** to become UNSET within the environment of the Broker Daemon.

Universal Automation Center Agent Customization

Universal Automation Center Agent Configuration

Configuration options for Universal Automation Center Agent (UAG) are stored in configuration file, **uags.conf**, in directory **`/etc/universal`** by default. See the Universal Automation Center Agent 7.2.x Reference Guide for details on configuring UAG.

UAG runs as a component managed by Universal Broker. UAG provides a component definition file, **uag**, that Universal Broker uses to start it and establish its runtime environment. **uag** is located in directory **`/etc/universal/comp`** by default.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

UAG uses the Universal Access Control List (UACL) configuration file as a level of product security. How UAG utilizes the UACL file is described in the Universal Agent 7.2.x User Guide.

Universal Command Manager Customization

Universal Command Manager Configuration

Configuration options for Universal Command Manager are stored in configuration file, **ucmd.conf**, in directory **`/etc/universal`** by default.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Manager.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Command Server Customization

Universal Command Server Configuration

Configuration options for Universal Command Server are stored in configuration file, **ucmds.conf**, in directory **`/etc/universal`** by default.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Server.

Universal Command Server runs as a component managed by Universal Broker. Universal Command Server provides a component definition file, **ucmd**, that Universal Broker uses to start the Server and establish its runtime environment. **ucmd** is located in directory **`/etc/universal/comp`** by default.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Command Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Command Server utilizes the UACL file is described in the Universal Agent 7.2.x User Guide.

Universal Connector Customization

Universal Connector Configuration

Configuration options for Universal Connector are stored in configuration file **usap.conf**, in directory **`/etc/universal`**, by default.

See the Universal Connector for SAP 7.2.x Reference Guide for details on configuring Universal Connector.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Connector SAP RFC Configuration

Universal Connector utilizes SAP's NW (Netweaver) RFC interface. The NW RFC interface uses configuration file **sapnwrfc.ini** to store information required for connecting to SAP systems. The **sapnwrfc.ini** file is an SAP resource that can be shared by multiple external tools using the SAP NW RFC interface. The **sapnwrfc.ini** file groups connection information into **destinations**. Each destination contains the connection information required to establish a connection to a particular SAP system.

Note

Universal Connector for AIX and Universal Connector for Solaris SPARC-based systems do not not require the SAP NW RFC libraries to run. The RFC interface for those version of Universal Connector use configuration file **sapnwrfc.ini** to store information required for connecting to SAP systems.

The connections defined in the **sapnwrfc.ini** file must be configured to meet your local SAP environment. A sample **sapnwrfc.ini** file is installed with Universal Connector to directory **`/opt/universal/usap/samp`**. The file is provided by SAP and contains complete documentation on its configuration.

In order for Universal Connector to find the configured **sapnwrfc.ini** file, it must be placed in the Universal Connector executable directory, or environment variable **RFC_INI** must be set to its full path name.

Universal Control Manager Customization

Universal Control Manager Configuration

Configuration options for Universal Control Manager are stored in configuration file, **uctl.conf**, in directory **/etc/universal** by default.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Manager.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Control Server Customization

Universal Control Server Configuration

Configuration options for Universal Control Server are stored in configuration file, **uctls.conf**, in directory **`/etc/universal`** by default.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Server.

Universal Control Server runs as a component managed by Universal Broker. Universal Control Server provides a component definition file, **uctl**, that Universal Broker uses to start the Server and establish its runtime environment. **uctl** is located in directory **/etc/universal/comp** by default

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Control Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Control Server utilizes the UACL file is described in the Universal Agent 7.2.x User Guide.

Universal Data Mover Manager Customization

Universal Data Mover Manager Configuration

Configuration options for UDM Manager are stored in configuration file, **udm.conf**, in directory **`/etc/universal`** by default.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring UDM Manager.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Data Mover Server Customization

Universal Data Mover Server Configuration

Configuration options for UDM Server are stored in configuration file, **udms.conf**, in directory **`/etc/universal`** by default.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring Universal Command Server.

UDM Server runs as a component managed by Universal Broker. UDM Server provides a component definition file, **udm**, that Universal Broker uses to start the Server and establish its runtime environment. **udm** is located in directory **`/etc/universal/comp`** by default

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

UDM Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Data Mover Server utilizes the UACL file is described in the Universal Agent 7.2.x User Guide.

Universal Event Monitor Manager Customization

Universal Event Monitor Manager Configuration

Configuration options for UEM Manager are stored in configuration file **uem.conf**, in directory **`/etc/universal`** by default.

See the Universal Event Monitor 7.2.x Reference Guide for details on configuring UEM Manager.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Event Monitor Server Customization

Universal Event Monitor Server Configuration

Configuration options for UEM Server are stored in configuration file, **uems.conf**, in directory **`/etc/universal`** by default.

See the [Universal Event Monitor 7.2.x Reference Guide] for details on configuring UEM Server.

UEM Server runs as a component managed by Universal Broker. UEM Server provides two component definition files, located in the default directory **`/etc/universal/comp`**, that Universal Broker uses to start the Server and establish its runtime environment:

- **uems** is used to start an event-driven UEM Server.
- **uemd** is used to start a demand-driven UEM Server.

The product executable files intended for command line use are located in the default directory **`/opt/universal/bin`**:

- **uem**
- **uemload**

This directory must be added to the PATH environment variable for intended users of the executable files.

UEM Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Event Monitor Server utilizes the UACL file is described in the Universal Agent 7.2.x User Guide.

Universal Query Customization

Universal Query Configuration

Configuration options for Universal Query are stored in configuration file, **uquery.conf**, in directory **`/etc/universal`** by default.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Query.

The product executable files intended for command line use are located in the default directory **/opt/universal/bin**. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Agent for UNIX - File Inventory Lists

- Universal Agent for UNIX - File Inventory Lists
- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Connector for SAP
- Universal Connector for PeopleSoft
- Universal Control Manager
- Universal Control Server
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Certificate
- Universal Copy
- Universal Encrypt
- Universal Message Translator
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- Universal Message Service (OMS)
- Universal Controller Command Line Interface (CLI)
- Universal Translate Tables (UTT)
- Python Distribution for Universal Agent
    - Provided Modules

Universal Agent for UNIX - File Inventory Lists

The Universal Agent installation for UNIX includes the files required for:

- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager and Server
- Universal Connector for PeopleSoft
- Universal Connector for SAP
- Universal Control Manager and Server
- Universal Data Mover Manager and Server
- Universal Event Monitor Manager and Server
- Universal Certificate
- Universal Copy
- Universal Encrypt
- Universal Message Translator
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- Universal Message Service (OMS)
- Universal Controller Command Line Interface (CLI)
- Python Distribution for Universal Agent

This page identifies the files installed with each of these Universal Agent components / utilities.

The file paths listed presume that the installation directory (**/opt**) and the variable file directory (**/var/opt**) are the defaults. These directories can be changed on most UNIX installations.

Universal Automation Center Agent

File	Description
/etc/universal/comp/uag	Universal Automation Center Agent component definition file.
/etc/universal/uags.conf	Universal Automation Center Agent configuration file.
/etc/universal/uftp.conf	Universal FTP Client configuration file.
/opt/universal/nls/uagmceng.umc	Universal Automation Center Agent message catalog.
/opt/universal/uagsrv/bin/ops_copyfile	Universal Automation Center Agent copy utility.

/opt/universal/uagsrv/bin/ops_monitor	Universal Agent file monitor program.
/opt/universal/uagsrv/bin/ops_renice	Universal Agent utility program to adjust process niceness.
/opt/universal/uagsrv/bin/ops_scan	Universal Automation Center Agent scan utility.
/opt/universal/uagsrv/bin/opsmerge.sh	Universal Agent configuration merge utility.
/opt/universal/uagsrv/bin/uagsrv	Universal Automation Center Agent component program.
/opt/universal/uagsrv/bin/uftp	Universal FTP Client program.
/opt/universal/uagsrv/samp	UAG sample directory.
/opt/universal/uagsrv/samp/OPSWISE-MIB.txt	MIB file.
/opt/universal/uagsrv/samp/parser.txt	Universal FTP Client file describing the format of file listings received from FTP servers, based on operating system and/or file system.
/opt/universal/uagsrv/uagsinst.src	Universal Automation Center Agent installation source file (included in Generic UNIX packaging only).
/var/opt/universal/uag	Universal Agent command Reference files directory
/var/opt/universal/trace	Universal Automation Center Agent trace file directory.

Universal Broker

File	Description
/etc/universal/comp	Component definition directory.
/etc/universal/SB_ConfigSetup.env	Paths to installed files.
/etc/universal/uacl.conf	Universal Access Control List configuration file.
/etc/universal/ubroker.conf	Universal Broker configuration file.
/opt/universal/nls/ussmceng.umc	Universal Broker message catalog.
/opt/universal/ubroker/bin/ubroker	Console broker program.
/opt/universal/ubroker/bin/ubrokerd	Daemon broker program.
/opt/universal/ubroker/tmpl/aclcfg	Template file for the Universal Access Control List (UACL) configuration.
/opt/universal/ubroker/tmpl/omscfg	Template file for the Universal Message Service (OMS) configuration.
/opt/universal/ubroker/tmpl/omscmp	Template file for the Universal Message Service (OMS) component definition.
/opt/universal/ubroker/tmpl/uagcfg	Template file for the Universal Automation Center Agent (UAG) configuration.
/opt/universal/ubroker/tmpl/uagcmp	Template file for the Universal Automation Center Agent (UAG) component definition.
/opt/universal/ubroker/tmpl/ubrcfg	Template file for the Universal Broker configuration.
/opt/universal/ubroker/tmpl/ucmcfg	Template file for the Universal Command Manager configuration.
/opt/universal/ubroker/tmpl/ucmcmp	Template file for the Universal Command Server component definition.
/opt/universal/ubroker/tmpl/ucscfg	Template file for the Universal Command Server configuration.
/opt/universal/ubroker/tmpl/uctcfg	Template file for the Universal Control configuration.
/opt/universal/ubroker/tmpl/uctcmp	Template file for the Universal Control Server component definition.
/opt/universal/ubroker/tmpl/udmcfg	Template file for the Universal Data Mover Manager configuration.
/opt/universal/ubroker/tmpl/udmcmp	Template file for the Universal Data Mover Server component definition.
/opt/universal/ubroker/tmpl/udscfg	Template file for the Universal Data Mover Server configuration.
/opt/universal/ubroker/tmpl/uedcmp	Template file for the Universal Event Monitor Demand-Driven Server component definition.

/opt/universal/ubroker/tmpl/uemcfg	Template file for the Universal Event Monitor Manager configuration.
/opt/universal/ubroker/tmpl/uemcmp	Template file for the Universal Event Monitor Event-Driven Server component definition.
/opt/universal/ubroker/tmpl/uescfg	Template file for the Universal Event Monitor Server configuration.
/opt/universal/ubroker/tmpl/uftcfg	Template file for the Universal FTP Client configuration.
/opt/universal/ubroker/tmpl/uppcfg	Template file for the Universal Connector for PeopleSoft configuration.
/opt/universal/ubroker/tmpl/uqrcfg	Template file for the Universal Query configuration.
/opt/universal/ubroker/tmpl/usapcfg	Template file for the Universal Connector for SAP configuration.
/opt/universal/ubroker/tmpl/utscfg	Template file for the Universal Control Server configuration.
/opt/universal/ubroker/samp	Broker sample directory.
/opt/universal/ubroker/ubrinst.src	Universal Broker installation source file (included in Generic UNIX packaging only).
/opt/universal/ubroker/ubrokerd	Broker daemon start script.
/var/opt/universal/cmdref	Universal Broker command Reference files directory.
/var/opt/universal/keystore	Broker Key Store.
/var/opt/universal/log	Broker message log directory.
/var/opt/universal/spool	Broker component database files.
/var/opt/universal/tmp	Universal Broker temporary directory.
/var/opt/universal/trace	Broker trace file directory.

Universal Command Manager

File	Description
/etc/universal/ucmd.conf	Universal Command Manager configuration file.
/opt/universal/bin/ucmd	Symbolic link to the Universal Command Manager program.
/opt/universal/nls/ucmmceng.umc	UCMD message catalog.
/opt/universal/ucmdmgr/bin/ucmd	Universal Command Manager program.
/opt/universal/ucmdmgr/bin/uencrypt	Universal Encrypt program file.
/opt/universal/ucmdmgr/samp	Manager sample file directory.
/opt/universal/ucmdmgr/ucmcinst.src	Manager installation source file (included in Generic UNIX packaging only).

Universal Command Server

File	Description
/opt/universal/bin/ucopy	Symbolic link to the Universal Copy program.
/opt/universal/bin/umet	Symbolic link to the Universal Message Translator program.
/etc/universal/comp/ucmd	Universal Control Server component definition file.
/etc/universal/ucmds.conf	Server configuration file.
/opt/universal/ucmdsrv/bin/ucmsrv	Server component program.
/opt/universal/ucmdsrv/bin/ucopy	Universal Copy utility used for binary file copies. Similar to the UNIX cat command.
/opt/universal/ucmdsrv/bin/umet	Universal Message Translator program.
/opt/universal/ucmdsrv/samp	Server sample directory.
/opt/universal/ucmdsrv/ucmsinst.src	Server installation source file (included in Generic UNIX packaging only).

/var/opt/universal/spool	Server component database files and spool file directory.
/var/opt/universal/trace	Server trace file directory.

Universal Connector for SAP

File	Description
/etc/universal/usap.conf	Server configuration file.
/opt/universal/bin/sapnwrfc.ini	Symbolic link to the SAP NW (NetWeaver) RFC interface `sapnwrfc.ini` configuration file in the **/opt/universal/usap/samp** directory.
/opt/universal/bin/usap	Symbolic link to the Universal Connector for SAP program.
/opt/universal/nls/uspmceng.umc	Universal Connector for SAP message catalog.
/opt/universal/usap/bin/usap	Universal Connector for SAP program.
/opt/universal/usap/samp	Server sample file directory. Note The **samp** directory contains the SAP NW (NetWeaver) RFC interface `sapnwrfc.ini` configuration file. However, for Universal Connector for AIX and Universal Connector for Solaris SPARC-based systems, **samp** contains SAP RFC interface `saprfc.ini` configuration file, as thise versions of Universal Connector do not require the SAP NW RFC libraries.
/opt/universal/usap/usapinst.src	Server installation source file (include in Generic UNIX packaging only).

Universal Connector for PeopleSoft

File	Description
/etc/universal/upps.conf	Manager configuration file.
/opt/universal/bin/upps	Symbolic link to the Universal Connector for PeopleSoft program.
/opt/universal/nls/uppmceng.umc	Universal Connector for PeopleSoft message catalog.
/opt/universal/upps/bin/upps	Universal Connector for PeopleSoft program.
/opt/universal/upps/samp	Server sample file directory.
/opt/universal/upps/uppsinst.src	Server installation source file (include in Generic UNIX packaging only).

Universal Control Manager

File	Description

/etc/universal/uctl.conf	Manager configuration file.
/opt/universal/bin/uctl	Symbolic link to the Manager program.
/opt/universal/nls/uctmceng.umc	Universal Control message catalog.
/opt/universal/uctlmgr/bin/uctl	Manager program.
/opt/universal/uctlmgr/samp	Manager sample file directory.
/opt/universal/uctlmgr/uctcinst.src	Manager installation source file (include in Generic UNIX packaging only).

Universal Control Server

File	Description
/etc/universal/comp/uctl	Universal Control Server component definition file.
/etc/universal/uctls.conf	Universal Control Server configuration file.
/opt/universal/uctlsrv/bin/uctsrv	Server component program.
/opt/universal/uctlsrv/samp	Server sample file directory.
/opt/universal/uctlsrv/uctsinst.src	Server installation source file (include in Generic UNIX packaging only).
/var/opt/universal/trace	Universal Control Server trace file directory.

Universal Data Mover Manager

File	Description
/etc/universal/udm.conf	Manager configuration file.
/opt/universal/bin/udm	Symbolic link to the Manager program.
/opt/universal/nls/udmmceng.umc	Universal Data Mover message catalog.
/opt/universal/udmmgr/bin/udm	Manager program.
/opt/universal/udmmgr/samp	Manager sample file directory.
/opt/universal/udmmgr/udmcinst.src	Manager installation source file (included in Generic UNIX packaging only).

Universal Data Mover Server

File	Description
/etc/universal/comp/udm	Universal Data Mover Server component definition file.
/etc/universal/udms.conf	Universal Data Mover Server configuration file.
/opt/universal/udmsrv/bin/udmsrv	Universal Data Mover Server program.
/opt/universal/udmsrv/samp	Universal Data Mover Server sample file directory.
/opt/universal/udmsrv/udmsinst.src	Universal Data Mover Server installation source file (include in Generic UNIX packaging only).
/var/opt/universal/trace	Universal Data Mover Server trace file directory.

Universal Event Monitor Manager

File	Description
/etc/universal/uem.conf	Manager configuration file.

/opt/universal/bin/uem	Symbolic link to the Manager program.
/opt/universal/nls/uemmceng.umc	Universal Event Monitor message catalog.
/opt/universal/uemmgr/bin/uem	Manager program.
/opt/universal/uemmgr/samp	Manager sample file directory.
/opt/universal/uemmgr/uemcinst.src	Manager installation source file (included in Generic UNIX packaging only).
/var/opt/universal/uem	

Universal Event Monitor Server

File	Description
/etc/universal/comp/uemd	Demand-driven Universal Event Monitor Server component definition file.
/etc/universal/comp/uems	Event-driven Universal Event Monitor Server component definition file.
/opt/universal/bin/uemload	Symbolic link to the UEM database load utility.
/etc/universal/uems.conf	Server configuration file.
/opt/universal/uemsrv/bin/uemload	UEM database load utility.
/opt/universal/uemsrv/bin/uemsrv	Server program.
/opt/universal/uemsrv/samp	Server sample file directory.
/opt/universal/uemsrv/uemsinst.src	Server installation source file (included in Generic UNIX packaging only).
/var/opt/universal/trace	Server trace file directory.

Universal Certificate

File	Description
/opt/universal/bin/ucert	Symbolic link to the Universal Certificate program.
/opt/universal/nls/ucrmceng.umc	Universal Certificate message catalog.
/opt/universal/ucert/bin/ucert	UCERT program
/opt/universal/ucert/ucrinst.src	Universal Certificate installation source file (included in Generic UNIX packaging only).

Universal Copy

File	Description
/opt/universal/nls/ucomceng.umc	Universal Copy message catalog.

Universal Encrypt

File	Description
/opt/universal/bin/uencrypt	Symbolic link to the Universal Encrypt program.
/opt/universal/nls/uenmceng.umc	Universal Encrypt message catalog

Universal Message Translator

File	Description

/opt/universal/nls/umemceng.umc	Universal Message Translator message catalog

Universal Products Install Merge

File	Description
/opt/universal/bin/upimerge	Symbolic link to the Universal Products Install Merge program.

Universal Query

File	Description
/etc/universal/uquery.conf	Universal Query configuration file.
/opt/universal/bin/uquery	Symbolic link to the Universal Query program.
/opt/universal/nls/uqrmceng.umc	Universal Query message catalog.
/opt/universal/uquery/bin/uquery	Universal Query program.
/opt/universal/uquery/samp	Universal Query sample file directory.
/opt/universal/uquery/uqryinst.src	Universal Query installation source file (included in Generic UNIX packaging only).

Universal Spool Utilities

File	Description
/opt/universal/nls/uslmceng.umc	Universal Spool Library message catalog.
/opt/universal/uspool/bin/uslist	Used to list the contents of Universal Spool files.
/opt/universal/uspool/bin/uslrm	Used to remove records from Universal Spool files.
/opt/universal/uspool/bin/udb_archive **/opt/universal/uspool/bin/udb_checkpoint** **/opt/universal/uspool/bin/udb_deadlock** **/opt/universal/uspool/bin/udb_dump** **/opt/universal/uspool/bin/udb_load** **/opt/universal/uspool/bin/udb_printlog** **/opt/universal/uspool/bin/udb_recover** **/opt/universal/uspool/bin/udb_stat** **/opt/universal/uspool/bin/udb_upgrade** **/opt/universal/uspool/bin/udb_verify**	Miscellaneous spool file utilities. Should be used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support.

Universal Message Service (OMS)

File	Description
/etc/universal/comp/oms	OMS Server component definition file.
/etc/universal/omss.conf	OMS Server configuration file.
/opt/universal/nls/omsmceng.umc	OMS message catalog.

/opt/universal/omssrv/bin/omsadm	OMS Administration utility program.
/opt/universal/omssrv/bin/omssrv	OMS Server program.
/opt/universal/omssrv/omssinst.src	OMS Server installation source file (included in Generic UNIX packaging only).
/opt/universal/omssrv/samp/oms.def	OMS Server sample component definition file.
/opt/universal/omssrv/samp/omss.conf	OMS Server sample configuration file.
/opt/universal/ubroker/tmpl/omscfg	Template file for the OMS Server configuration.
/opt/universal/ubroker/tmpl/omscmp	Template file for the OMS Server component definition.
/var/opt/universal/spool/oms	OMS Server component database files and spool file directory.

Universal Controller Command Line Interface (CLI)

File	Description
/etc/universal/cmdtools.props	Optional CLI configuration file.
/opt/universal/opscli/bin/ops-agent-status	Lists the status of one or more Agents.
/opt/universal/opscli/bin/ops-change-credential-password	Changes the runtime password for the specified Universal Controller credentials resource.
/opt/universal/opscli/bin/ops-change-user-password	Changes the password for the specified Universal Controller User account.
/opt/universal/opscli/bin/ops-connector-status	Lists the status of one or more Connectors.
/opt/universal/opscli/bin/ops-export-bulk	Performs a bulk export of all Controller database records.
/opt/universal/opscli/bin/ops-export-trigger	Performs an export of the specified triggers and any associated records.
/opt/universal/opscli/bin/ops-import-bulk	Imports Controller database records that were exported with ops-export-bulk.
/opt/universal/opscli/bin/ops-import-trigger	Imports triggers and associated records that were exported using ops-export-trigger.
/opt/universal/opscli/bin/ops-manual-setcompleted	Sets Manual task status to Success.
/opt/universal/opscli/bin/ops-manual-setstarted	Changes Manual task status from Action Required to Started.
/opt/universal/opscli/bin/ops-resume-agent	Allows a suspended Agent to submit tasks once again.
/opt/universal/opscli/bin/ops-resume-agent-cluster	Allows a suspended Agent Cluster to execute tasks once again.
/opt/universal/opscli/bin/ops-resume-agent-cluster-membership	Allows a specified Agent to rejoin specified Agent Cluster.
/opt/universal/opscli/bin/ops-set-agent-cluster-task-execution-limit	Sets the task execution limit for an Agent Cluster.
/opt/universal/opscli/bin/ops-set-agent-task-execution-limit	Sets the task execution limit for an Agent.
/opt/universal/opscli/bin/ops-suspend-agent	Temporarily prevents a specified Agent from submitting tasks.
/opt/universal/opscli/bin/ops-suspend-agent-cluster	Temporarily prevents the Agents in the specified Cluster from submitting tasks.
/opt/universal/opscli/bin/ops-suspend-agent-cluster-membership	Temporarily removes an Agent from cluster membership.
/opt/universal/opscli/bin/ops-task-cancel	Cancels a task.
/opt/universal/opscli/bin/ops-task-forcefinish	Force Finishes a task.
/opt/universal/opscli/bin/ops-task-hold	Places a task on hold.
/opt/universal/opscli/bin/ops-task-launch	Executes a task.
/opt/universal/opscli/bin/ops-task-list	Lists the specified tasks.
/opt/universal/opscli/bin/ops-task-release	Releases a held task.
/opt/universal/opscli/bin/ops-task-rerun	Re-executes the specified task.

/opt/universal/opscli/bin/ops-task-setpriority	Changes the execution priority of a Started task.
/opt/universal/opscli/bin/ops-task-skip	Skips the specified task instance.
/opt/universal/opscli/bin/ops-task-status	Displays the status of the task instance(s) associated with a task.
/opt/universal/opscli/bin/ops-trigger-disable	Disables the specified trigger(s).
/opt/universal/opscli/bin/ops-trigger-enable	Enables the specified trigger(s).
/opt/universal/opscli/bin/ops-trigger-now	Satisfies all conditions in the specified trigger and launches its associated tasks.
/opt/universal/opscli/bin/ops-trigger-status	Displays the status of the specified trigger(s).
/opt/universal/opscli/bin/ops-update-resource-limit	Sets the resource limit for a resource.
/opt/universal/opscli/bin/ops-variable-list	Displays the specified variable's value.
/opt/universal/opscli/bin/ops-variable-set	Sets the specified variable's value.
/opt/universal/opscli/bin/uagcmd	CLI executable command.
/opt/universal/opscli/samp	CLI sample directory.

Universal Translate Tables (UTT)

Code page files used for text translation between different operating systems and product message catalogs.

File
/opt/universal/nls/default.stt
/opt/universal/nls/raw.stt
/opt/universal/nls/iso8859-1.utt
/opt/universal/nls/iso8859-2.utt
/opt/universal/nls/iso8859-3.utt
/opt/universal/nls/iso8859-4.utt
/opt/universal/nls/iso8859-5.utt
/opt/universal/nls/iso8859-6.utt
/opt/universal/nls/iso8859-7.utt
/opt/universal/nls/iso8859-8.utt
/opt/universal/nls/iso8859-9.utt
/opt/universal/nls/iso8859-10.utt
/opt/universal/nls/iso8859-13.utt
/opt/universal/nls/iso8859-14.utt
/opt/universal/nls/iso8859-15.utt

Python Distribution for Universal Agent

Installed under **/opt/universal/python**.

The Python distribution provided by the Universal Agent install is a fully-functional Python distribution that has complete support for Python script execution, virtual environment creation, and/or add-on module installations via pip.

When installing new Python modules to this distribution’s install directory, remember to use the provided pip binary to ensure that the module is installed to the correct path.

Starting with the Universal Agent 6.8.0.0 release, the path to our custom Python distribution will no longer contain version numbers. This will make it possible to upgrade the version of Python included with the Universal Agent package without affecting scripts that take advantage of the version-independent path.

For example, when you add `#!/opt/universal/python/bin/python` to the start of your Python script, you will be protected against future version changes that the Universal Agent install may provide.

Note

To preserve compatibility with existing Stonebranch marketplace scripts, the Universal Agent install also provides an `/opt/universal/python3.6/bin` directory. This directory contains a python3 symbolic link that points to the current Python binary installed to `/opt/universal/python/bin`. This will allow scripts using `#!/opt/universal/python3.6/bin/python3` to work with the latest install.

It also means that the Python version reported using this path may no longer be 3.6 (starting with Universal Agent 6.8.0.0, it will be 3.7). Support for the python3.6 symlink may go away at any time, so protect any new scripts by pointing them to the version-independent `/opt/universal/python/bin/python` path.

Provided Modules

The modules provided by the Universal Agent's Python distribution are listed below. Where possible, the dependencies for each of these modules are also provided.

Stonebranch does not own, support, or maintain any of these modules. The link for each module will direct you to that project's homepage.

Module	Version	Notes
azure-storage-blob	12.7.1	12.1.1 for AIX and Solaris SPARC
azure-storage-logging	0.5.1	AIX, Solaris SPARC only
botocore	1.14.8	
boto3	1.11.8	
docker	4.1.0	
google-cloud-storage	1.25.0	
paramiko	2.7.1	
requests	2.22.0	
hdbcli	2.7.23	Linux Intel only
pyodbc	4.0.30	Linux Intel only
cx-Oracle	8.0.1	Linux Intel only
hdfs	2.5.8	Linux only
kubernetes	12.0.1	Linux only

Universal Agent for SOA for UNIX Installation

Currently, UNIX runs the following version of Universal Agent for SOA:

- **6.7.0.0**

These pages provide information for that version.

Overview

The following information is provided for the installation of Universal Agent for SOA for UNIX:

- Installation Package
- Installation Requirements
- Pre-Installation-Upgrade Backups
- Distribution File
- Deployment Options
- Installation Procedures
- File Inventory Lists

Licensing

Universal Agent for SOA allows operation by connector and endpoint count, based on the license information it receives from the UAC Server component. (See your Stonebranch, Inc. account representative for details.)

(For licensing information, see UNIX Installation - Licensing.)

Universal Agent for SOA for UNIX - Installation Package

Package Components

The Universal Agent for SOA 6.7.x for UNIX package includes the following product components:

- Universal Application Container Server
- Universal Application Container
- Universal Application Interface

Component Compatibility

The following table identifies the compatibility of Universal Agent for SOA for UNIX 6.7.x with previous component / product versions.

Component	Compatibility
Universal Agent for SOA 6.7.x	Universal Command Manager 6.7.0, 6.6.0, 6.5.0, 6.4.0, 6.3.0, 6.2.0, 5.2.0, 5.1.0, 4.3.0, 4.2.0, 4.1.0, and 3.2.0.

The component references pertain to all supported platforms for that version.

Universal Agent for SOA for UNIX - Installation Requirements

UNIX Versions

To install Universal Agent for SOA for UNIX, you must have a Linux environment that satisfies the following requirements:

- RedHat Package Manager (RPM) distributions
  - Intel x86_64-based system
  - Linux Kernel 3.10 or greater
  - gllibc 2.17-78 or higher
- Debian distributions
  - Linux Kernel 3.0 or greater
  - glibc 2.15 or higher

Additional Requirements

In addition, you must have:

- 512MB RAM minimum, 1 GB or more preferred.
- 150 MB free disk space.
- TCP/IP socket implementation.
- Superuser (root) access.
- Universal Agent 6.3.0.1 or later.

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x to make sure that your platform is supported before performing an installation.

Universal Agent for SOA for UNIX - Pre-Installation-Upgrade Backups

Universal Agent for SOA for UNIX - Pre-Installation / Upgrade Backups

The installation process overwrites the current files (exception: see the Note for Log4jConfiguration.xml in the following table), effectively removing your modifications. Backing up these files will optimize the time it takes you to get up and running after installing or upgrading.

The following list identifies the files - and their locations - that should be backed up or copied before you install a new release or upgrade a current release.

File	Location
UAC.xml File	**`/etc/universal`**
Log4jConfiguration.xml File	**`/etc/universal`** (UAC) **`/etc/universal`** (UAI) Note The installation process does not overwrite Log4jConfiguration.xml files.
JMS Provider Client Jar Files	**`/opt/universal/uac/container/webapps/axis2/WEB-INF/lib`** Note The names of the jar files differ depending on which JMS Provider you are using.
JMS Provider Client Properties Files	**`/opt/universal/uai/xml`** Note These are suggested locations; you can place these files anywhere on the files system. If you have located these files under the **uai** directory, you should back them up.
Payload Files	Normally, payload files should be located on the host system where Universal Command Manager is installed. If you have chosen to store them elsewhere, the suggested location is: **`/opt/universal/uai/xml`** Note You can place these files anywhere on the files system. If you have located these files under the **uai** directory, you should back them up.

Universal Agent for SOA for UNIX - Distribution File

- UNIX Distribution File
- Obtaining the Distribution File
- Distribution File Format

UNIX Distribution File

Stonebranch, Inc. provides different Universal Agent for SOA for UNIX packages for different types of UNIX operating systems.

Obtaining the Distribution File

To obtain the Universal Agent for SOA for UNIX package for your type of UNIX operating system, you must download the corresponding product distribution file from the Stonebranch Customer Portal.

Note

A customer user name and password — provided by Stonebranch, Inc. — are required to access this area.

After a distribution file has been downloaded, the installation files contained in that file must be extracted before the product can be installed (see Universal Agent for SOA for UNIX - Installation Procedures).

Distribution File Format

The name of each Universal Agent for SOA for UNIX distribution file has the following format:

`sb-soa-Version.Release.Modification Level.Maintenance Level-operating system-version(.release)(-platform).tar.Z`

For example: **`sb-soa-6.7.x.x-linux-2.4-i386.tar.Z`**

In this format:

- **Version** is the current version of Universal Agent for SOA.
- **Release** is the current release of Universal Agent for SOA.
- **Modification Level** is the current Universal Agent feature set.
- **Maintenance Level** is the Universal Agent build level.
- **operating system** is the name of the operating system (for example, AIX or Linux).
- **version(.release)** is the supported version and, optionally, the release of the operating system.
- **platform** is the targeted hardware platform (for example, i386). It is included in the file name only if there is more than one platform available for the specified operating system.

Universal Agent for SOA for UNIX - Deployment Options

- Deployment Options
- Single-Server Deployment (SSD)
  - SSD Deployment Flow of Events
- Distributed Server Deployment (DSD)
  - DSD Deployment Flow of Events

Deployment Options

Deployment of Universal Agent for SOA has been designed to be flexible in order to fit the needs of your Enterprise IT.

There are two main deployment options:

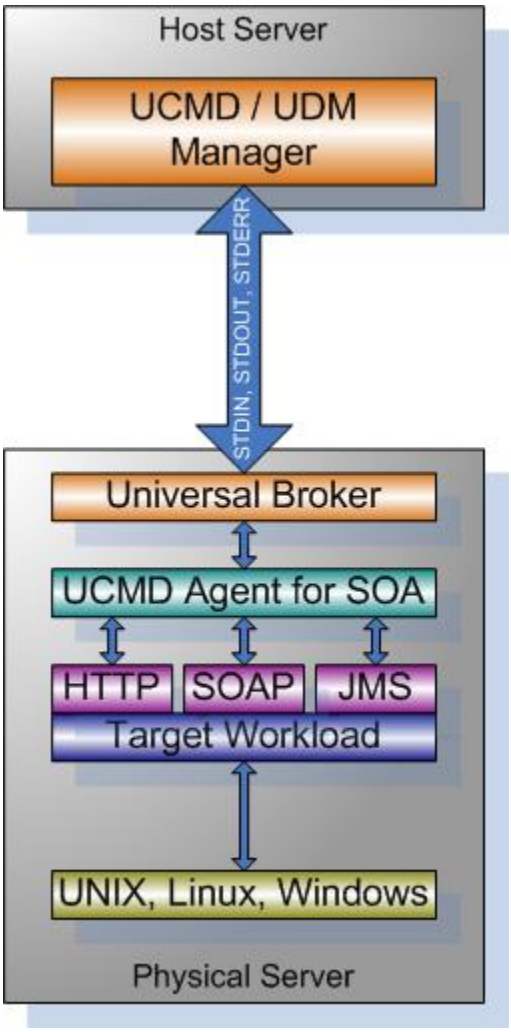
1. Single-Server Deployment (SSD)
2. Distributed Server Deployment (DSD)

SSD is the default deployment.

Single-Server Deployment (SSD)

Single-Server Deployment (SSD) is where all components, with the exception of the Universal Command Agent managers, are deployed to the same physical server. This includes Universal Command Agent for SOA, Universal Broker and associated components, and your target workload(s).

The following figure illustrates the Single Server Deployment (SSD).



SSD Deployment Flow of Events

The following list identifies the flow of events that occur with SSD deployment:

1. The calling application invokes the Universal Command (UCMD) Manager to execute a remote workload. For this example the remote workload, or target workload, is an internet or message based workload.

   Note

   Although this figure implies the use of a Java-based workload, you can execute any remote workload, regardless of what language the workload is implemented in, as long as it has an interface that supports HTTP, SOAP, or JMS.

2. The request is forwarded to the Universal Broker specified in the invocation of the Universal Command Manager, which then spawns the UCMD Server and passes the workload execution request to Universal Command Agent for SOA.
3. Universal Command Agent for SOA will execute the workload deployed on the same physical server and return any messages or data back to the Universal Command Manager.

Distributed Server Deployment (DSD)

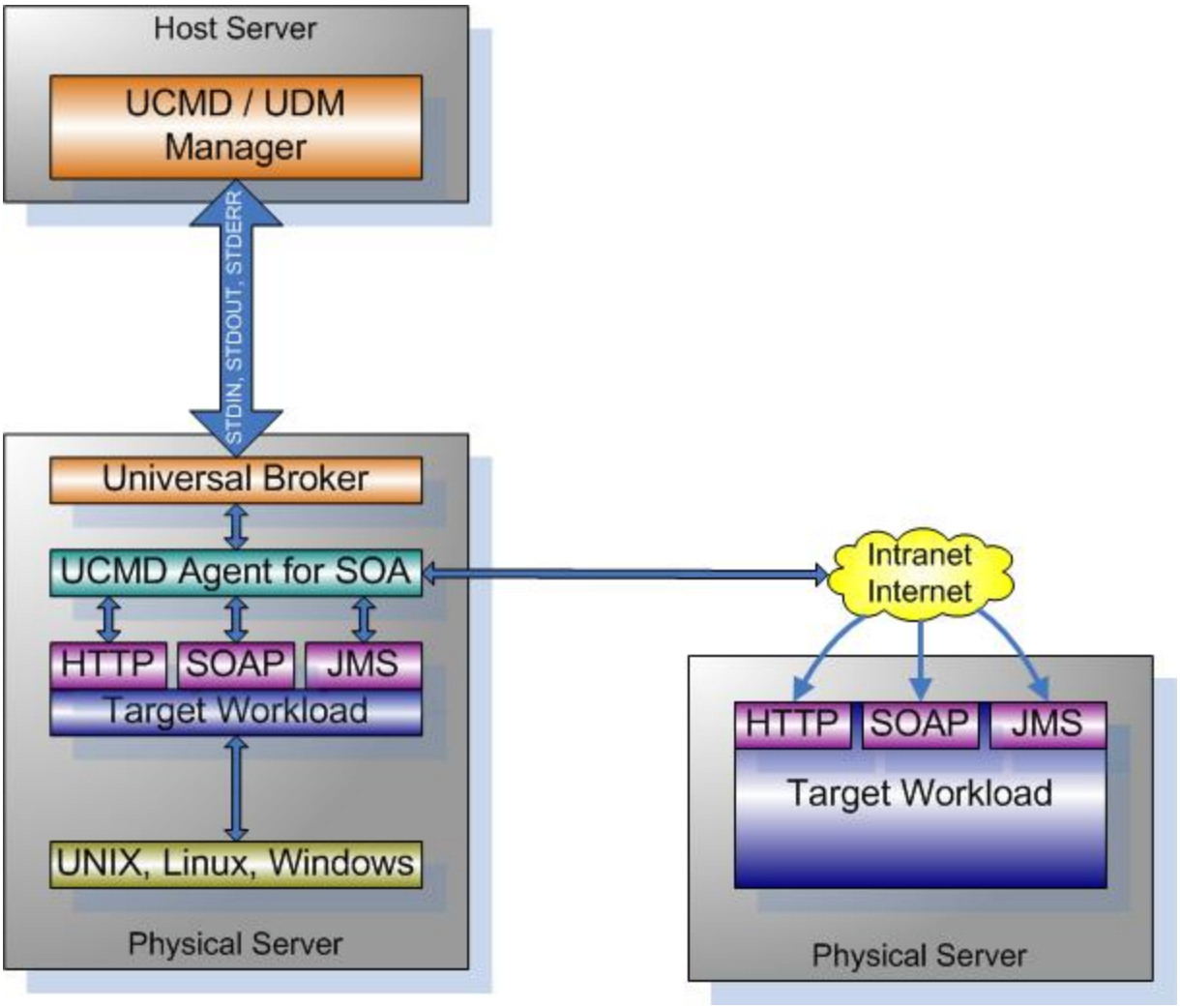
Distributed Server Deployment (DSD) is where:

- Universal Broker and Universal Command Agent for SOA are located together on one physical server.
- Target workload is located on a different physical server.

Note

The target workload location is user-defined. The SERVICE_URL option specifies how Universal Command Agent for SOA knows where to look for the target workload.

The following figure illustrates the Distributed Server Deployment (DSD).



DSD Deployment Flow of Events

The flow is nearly the same as with the Single Server Deployment except that the location of the target workload is not **localhost**.

Universal Agent for SOA for UNIX - Installation Procedures

Universal Agent for SOA for UNIX Installation Procedures

The following procedures are provided for the installation and modification of Universal Agent for SOA for UNIX:

- Universal Agent for SOA for Linux Installation

Installation Process

Installation is a straightforward process using the platform-specific package. The installed environment is self-contained and follows current Universal Agent installation standards.

Installation Prerequisite

Before installing Universal Command Agent for SOA 6.7.x, you must install Universal Products 3.2.0 or later.

Universal Agent for SOA for Linux Installation

- Product Distribution File
- Unpacking and Installation Procedures
- Removing Universal Agent for SOA for Linux
- Listing Universal Agent for SOA for Linux Information

Product Distribution File

Universal Agent for SOA 6.7.0.x is available as a Debian installation package.

The name of the Universal Agent for SOA for Linux distribution file has the following format:

sb-soa-*v.r.m.l*-linux-*arch*[-*pkg*].tar.Z

In this format:

- *v.r.m.l* is the version, release, modification, and level of the Universal Agent for SOA application.
- *arch* is the base target architecture.
- *pkg* is an optional package type identifier.

For example, the name of the 64-bit Debian distribution for the 6.7.0.0 release that supports Linux kernel versions 3 and higher is **sb-soa-6.7.0.0-linux-3-x86_64-deb.tar.Z**.

Unpacking and Installation Procedures

To unpack and install Universal Agent for SOA, perform the following steps:

Step 1	Create a directory (or select an existing directory) in which to save the package file.
Step 2	Save the package file into that directory.

Step 3	Uncompress and extract the installation file in the current working directory. The command to extract the files is:
	Debian Package
	`zcat sb-soa-6.7.x.x-linux-3-x86_64-deb.tar.Z | tar xvf -`
	RPM Package
	`zcat sb-soa-6.7.x.x-linux-2.6-x86_64.tar.Z | tar xvf -`
	You also can use the **gzip** command to uncompress and extract the files:
	Debian Package
	`gzip -cd sb-soa-6.7.x.x-linux-3-x86_64-deb.tar.Z | tar xvf -`
	RPM Package
	`gzip -cd sb-soa-6.7.x.x-linux-2.6-x86_64.tar.Z | tar xvf -`

Step 4	After the extraction is complete, run the installation script, upsinst, which executes the debian packaging commands: `./upsinst` An installation log is written to file **install.log** in the current directory. **upsinst** automatically restarts the Universal Broker daemon, **ubrokerd**, at the end of the install.
Step 5	From the license file that was sent to you by Stonebranch, Inc., add the license information to the **`/etc/universal/uacs.conf`** file.

Step 6	Recycle Universal Broker using the following commands (cd to **`/opt/universal/ubroker`**):
	
	First:
	
	`./ubrokerd stop`
	
	Then:
	
	`./ubrokerd start`

Step 7	Use Universal Query (cd to **`/opt/universal/bin`**) to validate that the Universal Application Container Server component of Universal Agent for SOA 6.7.x is running:
	
	`uquery -i localhost` (or the name of your server)
	
	The output for Universal Application Container Server should have the following format:
	
	`Component ID.............: 1360109684`
	`Component Name...........: uac (Server)`
	`Component Description....: Universal Application Container Server`
	`Component Version........: 6.7.0 Level 0 Release Build 100`
	`Component Type...........: uac`
	`Component Process ID.....: 23331000`
	`Component Start Time.....: 18:14:42`
	`Component Start Date.....: 01/05/19`
	`Component Command ID.....: uac`
	`Component State..........: REGISTERED`
	`Component MGR UID........:`
	`Component MGR Work ID.....:`
	`Component MGR Host Name...:`
	`Component MGR IP Address..:`
	`Component MGR Port.......:`
	`Component Comm State.....: ESTABLISHED`
	`Component Comm State Time.: 18:14:44`
	`Component Comm State Date.: 01/05/19`
	`Component MGR Restartable.: NO`
	`Component Comment........:`

Removing Universal Agent for SOA for Linux

Note

Before removing Universal Agent for SOA for Linux, stop the **ubrokerd** daemon. Also, it is strongly recommended that you back up existing data before removing Universal Agent for SOA for Linux.

Debian Package
Universal Agent for SOA for Linux is removed with the **dpkg** command. The command must be executed with the superuser authority.

```
sudo dpkg -r ups
```

RPM Package
Universal Agent for SOA for Linux is removed with the **rpm** command. The command must be executed with the superuser authority.

```
sudo rpm -e ups
```

Listing Universal Agent for SOA for Linux Information

Debian Package
To list information for Universal Agent for SOA for Linux, issue the following command:

```
dpkg -l ups
```

RPM Package
To list information for Universal Agent for SOA for Linux, issue the following command:

```
rpm -qi ups
```

Universal Agent for SOA for UNIX - File Inventory Lists

- Overview
- Parent Directories
- Product Directories and Files
  - /opt/universal Parent Directory
  - /var/opt/universal Parent Directory
  - /etc/universal Parent Directory
  - /opt/universal/jre Directory

Overview

This page identifies the Universal Agent for SOA file system hierarchy and its contents.

The parent directories under which Universal Agent for SOA operates are based on the existingUniversal Agent deployment. There should be no product components in these directories, just the product directories.

The product directories contain the product components either directly or in sub directories and are divided into two categories: runtime and logging.

- Runtime directories are for runtime operation of the product; they are not written to.
- Logging directories are where database components, audit and logging files reside, all components that are written to.

Parent Directories

For UNIX, the parent directories are:

- **/opt/universal**
- **/var/opt/universal**

The following sections identify the directories and files located under each of these parent directories.

Product Directories and Files

/opt/universal Parent Directory

The following table identifies the Universal Agent for SOA for UNIX product directories and files located under the **/opt/universal** parent directory.

Directory / File	Sub-Directory / File	Description
uac		Directory containing artifacts for the UAC (Universal Application Container) component. It includes the following sub-directories and files.
	bin	Sub-directory containing the **uacsrv** executable. It requires no user interaction.
	lib	Sub-directory containing the **uacsrv.jar** file needed for communication between the **uacsrv** and **uac** components. It requires no user interaction.
	container	Sub-directory containing the libraries and other deployable objects needed for UAC operation. The only reason to explore this directory is if you are running the JMS Connector and need to deploy your JMS provider client jar files.
	shutdownUAC.sh	Shutdown script for UAC. You usually will not need to use this script, as it is the responsibility of Universal Broker to shut down UAC.
	startUAC.sh	Start-up script for UAC. You usually will not need to use this script, as it is the responsibility of Universal Broker to start UAC.
	uacValidateInbound.sh	Script that validates the contents of the UAC.xml file using the schema definition contained in **UAC.xsd**.
	UAC.xsd	File used by UAC to validate content of the **UAC.xml** inbound configuration file. It requires no user interaction.
uai		Directory containing the artifacts for the UAI (Universal Application Interface) component. It includes the following sub-directories and files.
	lib	Sub-directory containing the libraries needed for UAI operation. It usually requires no user interaction.
	UAI.sh	Shell script used to invoke the UAI component. It usually requires no user interaction.

jre		Directory containing the Open Java Runtime Environment for the UAC, UAI, and UAC Server components. It usually requires no user interaction:
		
		• %dir /opt/universal/jre
		  • %doc /opt/universal/jre/ASSEMBLY_EXCEPTION
		  • %doc /opt/universal/jre/LICENSE
		  • %doc /opt/universal/jre/THIRD_PARTY_README

- %dir /opt/universal/jre/bin
  - /opt/universal/jre/bin/clhsdb
  - /opt/universal/jre/bin/hsdb
  - /opt/universal/jre/bin/java
  - /opt/universal/jre/bin/jjs
  - /opt/universal/jre/bin/keytool
  - /opt/universal/jre/bin/orbd
  - /opt/universal/jre/bin/pack200
  - /opt/universal/jre/bin/policytool
  - /opt/universal/jre/bin/rmid
  - /opt/universal/jre/bin/rmiregistry
  - /opt/universal/jre/bin/servertool
  - /opt/universal/jre/bin/tnameserv
  - /opt/universal/jre/bin/unpack200
- %dir /opt/universal/jre/lib
  - /opt/universal/jre/lib/calendars.properties
  - /opt/universal/jre/lib/charsets.jar
  - /opt/universal/jre/lib/classlist
  - /opt/universal/jre/lib/content-types.properties
  - /opt/universal/jre/lib/currency.data
  - /opt/universal/jre/lib/flavormap.properties
  - /opt/universal/jre/lib/hijrah-config-umalqura.properties
  - /opt/universal/jre/lib/jce.jar
  - /opt/universal/jre/lib/jexec
  - /opt/universal/jre/lib/jsse.jar
  - /opt/universal/jre/lib/jvm.hprof.txt
  - /opt/universal/jre/lib/logging.properties
  - /opt/universal/jre/lib/management-agent.jar
  - /opt/universal/jre/lib/meta-index
  - /opt/universal/jre/lib/net.properties
  - /opt/universal/jre/lib/psfontj2d.properties
  - /opt/universal/jre/lib/psfont.properties.ja
  - /opt/universal/jre/lib/resources.jar
  - /opt/universal/jre/lib/rt.jar
  - /opt/universal/jre/lib/sound.properties
  - /opt/universal/jre/lib/tzdb.dat
- %dir /opt/universal/jre/lib/amd64
  - /opt/universal/jre/lib/amd64/jvm.cfg
  - /opt/universal/jre/lib/amd64/libattach.so
  - /opt/universal/jre/lib/amd64/libawt_headless.so
  - /opt/universal/jre/lib/amd64/libawt.so
  - /opt/universal/jre/lib/amd64/libawt_xawt.so
  - /opt/universal/jre/lib/amd64/libdt_socket.so
  - /opt/universal/jre/lib/amd64/libfontmanager.so
  - /opt/universal/jre/lib/amd64/libhprof.so
  - /opt/universal/jre/lib/amd64/libinstrument.so
  - /opt/universal/jre/lib/amd64/libj2gss.so
  - /opt/universal/jre/lib/amd64/libj2pcsc.so
  - /opt/universal/jre/lib/amd64/libj2pkcs11.so
  - /opt/universal/jre/lib/amd64/libjaas_unix.so
  - /opt/universal/jre/lib/amd64/libjava_crw_demo.so
  - /opt/universal/jre/lib/amd64/libjava.so
  - /opt/universal/jre/lib/amd64/libjawt.so
  - /opt/universal/jre/lib/amd64/libjdwp.so
  - /opt/universal/jre/lib/amd64/libjpeg.so
  - /opt/universal/jre/lib/amd64/libjsdt.so
  - /opt/universal/jre/lib/amd64/libjsig.so
  - /opt/universal/jre/lib/amd64/libjsoundalsa.so
  - /opt/universal/jre/lib/amd64/libjsound.so
  - /opt/universal/jre/lib/amd64/liblcms.so
  - /opt/universal/jre/lib/amd64/libmanagement.so
  - /opt/universal/jre/lib/amd64/libmlib_image.so
  - /opt/universal/jre/lib/amd64/libnet.so
  - /opt/universal/jre/lib/amd64/libnio.so
  - /opt/universal/jre/lib/amd64/libnpt.so
  - /opt/universal/jre/lib/amd64/libsaproc.so
  - /opt/universal/jre/lib/amd64/libsctp.so
  - /opt/universal/jre/lib/amd64/libsplashscreen.so
  - /opt/universal/jre/lib/amd64/libsunec.so
  - /opt/universal/jre/lib/amd64/libunpack.so
  - /opt/universal/jre/lib/amd64/libverify.so
  - /opt/universal/jre/lib/amd64/libzip.so
- %dir /opt/universal/jre/lib/amd64/jli
  - /opt/universal/jre/lib/amd64/jli/libjli.so
- %dir /opt/universal/jre/lib/amd64/server
  - /opt/universal/jre/lib/amd64/server/libjsig.so
  - /opt/universal/jre/lib/amd64/server/libjvm.so
  - /opt/universal/jre/lib/amd64/server/Xusage.txt
- %dir /opt/universal/jre/lib/applet
- %dir /opt/universal/jre/lib/cmm
  - /opt/universal/jre/lib/cmm/CIEXYZ.pf
  - /opt/universal/jre/lib/cmm/GRAY.pf
  - /opt/universal/jre/lib/cmm/LINEAR_RGB.pf
  - /opt/universal/jre/lib/cmm/PYCC.pf
  - /opt/universal/jre/lib/cmm/sRGB.pf
- %dir /opt/universal/jre/lib/ext
  - /opt/universal/jre/lib/ext/cldrdata.jar
  - /opt/universal/jre/lib/ext/dnsns.jar
  - /opt/universal/jre/lib/ext/jaccess.jar
  - /opt/universal/jre/lib/ext/localedata.jar
  - /opt/universal/jre/lib/ext/meta-index
  - /opt/universal/jre/lib/ext/nashorn.jar
  - /opt/universal/jre/lib/ext/sunec.jar
  - /opt/universal/jre/lib/ext/sunjce_provider.jar
  - /opt/universal/jre/lib/ext/sunpkcs11.jar
  - /opt/universal/jre/lib/ext/zipfs.jar
- %dir /opt/universal/jre/lib/images

		• %dir /opt/universal/jre/lib/images/cursors
		  • /opt/universal/jre/lib/images/cursors/cursors.properties
		  • /opt/universal/jre/lib/images/cursors/invalid32x32.gif
		  • /opt/universal/jre/lib/images/cursors/motif_CopyDrop32x32.gif
		  • /opt/universal/jre/lib/images/cursors/motif_CopyNoDrop32x32.gif
		  • /opt/universal/jre/lib/images/cursors/motif_LinkDrop32x32.gif
		  • /opt/universal/jre/lib/images/cursors/motif_LinkNoDrop32x32.gif
		  • /opt/universal/jre/lib/images/cursors/motif_MoveDrop32x32.gif
		  • /opt/universal/jre/lib/images/cursors/motif_MoveNoDrop32x32.gif
		• %dir /opt/universal/jre/lib/management
		  • /opt/universal/jre/lib/management/jmxremote.access
		  • /opt/universal/jre/lib/management/jmxremote.password.template
		  • /opt/universal/jre/lib/management/management.properties
		  • /opt/universal/jre/lib/management/snmp.acl.template
		• %dir /opt/universal/jre/lib/security
		  • /opt/universal/jre/lib/security/blacklisted.certs
		  • /opt/universal/jre/lib/security/cacerts
		  • /opt/universal/jre/lib/security/java.policy
		  • /opt/universal/jre/lib/security/java.security
		• %dir /opt/universal/jre/lib/security/policy
		• %dir /opt/universal/jre/lib/security/policy/limited
		  • ** /opt/universal/jre/lib/security/policy/limited/local_policy.jar
		  • /opt/universal/jre/lib/security/policy/limited/US_export_policy.jar
		• %dir /opt/universal/jre/lib/security/policy/unlimited
		  • /opt/universal/jre/lib/security/policy/unlimited/local_policy.jar
		  • /opt/universal/jre/lib/security/policy/unlimited/US_export_policy.jar

nls/uacmceng.umc		Universal Application Container Server English message catalog.
Readme.ups		Universal Agent for SOA for UNIX installation README file.
ubroker/tmpl/uaccfg		Universal Application Container Server configuration template file.
ubroker/tmpl/uaccmp		Universal Application Container Server component definition template file.

/var/opt/universal Parent Directory

The following table identifies the Universal Agent for SOA for UNIX product directories and files located under the **/var/opt/universal** parent directory.

Directory	Sub-Directory / File	Description
log/uac		Directory containing logging and work artifacts for the UAC component. It includes the following sub-directories and files.
	temp	Sub-directory used by UAC for temporary file persistence. It requires no user interaction.
	work	Sub-directory used by the web services framework for temporary operations. It requires no user interaction.
	catalina.out	Container engine log file.
	container.log	Web services framework log file.
	derby.log	Database log file.
	uac.log	UAC log file.
log/uai		Directory containing the logging and work artifacts for the UAI component. It is empty by default. If **trace** or **info** is enabled, the output would be written to this directory.
uac_db		Directory containing the database components, log, property, and lock files. It requires no user interaction.

/etc/universal Parent Directory

The following table identifies the Universal Agent for SOA for UNIX files located under the **/etc/universal** parent directory.

File	Description
uac_log4jConfiguration.xml	Logger configuration file that you may need to modify if you want to change the logging level of the UAC component.
uai_log4jConfiguration.xml	Logger configuration file that you may need to modify if you want to change the logging level of the UAI component.
uacs.conf	Universal Application Container Server configuration file.
UAC.xml	Inbound operation configuration file that you may need to modify to support your inbound workload operations.
comp/uac	Universal Application Container Server component definition file.

/opt/universal/jre Directory

The following table identifies the third-party Open Java Runtime Environment files packaged with the Universal Agent for SOA for UNIX. The files are located under the **/opt/universal/jre** directory.

File	Description
bin	Open JRE binary executable files.
lib	Open JRE Library files.
LICENSE	Open JRE License README file.
THIRD_PARTY_README	Open JRE Third Party README file.
ASSEMBLY_EXCEPTION	Open JRE Assembly Exception License README file.

UNIX Installation - Licensing

- Licensing Universal Agent for UNIX Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

Licensing Universal Agent for UNIX Components

After Universal Agent for UNIX has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Connector for SAP
- Universal Connector for PeopleSoft
- Universal Application Container Server

Note

Universal Application Container Server (as a component of Universal Command Agent for SOA 6.4.x for UNIX) is packaged, and licensed, separately.

Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

Format

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: **Indesca_Stonebranch_UNIX_A1.txt**

- For Universal Command Manager, **Indesca** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.
- For Universal Data Mover Manager, **Infitran** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.

Sample

The following is a sample Universal Command Manager for UNIX product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "UNIX"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Entering License Information

Enter each component's product license file information into its configuration file:

- Universal Command Manager: **ucmd.conf**
- Universal Data Mover Manager: **udm.conf**
- Universal Connector for SAP: **usap.conf**
- Universal Connector for PeopleSoft: **upps.conf**

- Universal Application Container Server: **uacs.conf**

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it:

Stop Universal Broker	`ubrokerd stop`
Start Universal Broker	`ubrokerd start`

IBM i Installation

- Introduction
- Installation Summary
- Naming Conventions
- Detailed Information

Introduction

These pages provide information on the installation of Stonebranch Inc's Workload Automation 5 on the IBM i operating system. Unless otherwise specified, all references to Workload Automation 5 for IBM i in these pages refer to version 5.1.1.

All Workload Automation 5 for IBM i components are provided in an easily installed, one-time installation package.

Note

Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Workload Automation 5 component is running, including manager components. The local Broker maintains product configuration data for all components that have a configuration file.

Installation Summary

Warning

This installation procedure is intended only for default installations into simple environments. Installing into high availability environments may require a customized installation. For customized installation, parallel installation or multiple system roll out see the Installation Guide for IBM i.

Step 1	Download the distribution file from the Stonebranch Customer Portal to a computer running a supported version of Windows or UNIX.
Step 2	Extract the Licensed Program Save File from the distribution file.
Step 3	If transferring the installation file to the native, QSYS file system, create a save file, UNV511, on the target IBM I system in library QGPL. This step is not necessary if transferring the file to the IFS.
Step 4	FTP the Licensed Program Save File to either the save file created in step 1 or to /QSYS.LIB/QGPL.LIB/UNV511.SAVF.
Step 5	Execute the command RSTLICPGM to install Workload Automation 5 to the default libraries. Refer to Installation Procedures for additional parameters for the RSTLICPGM command.
Step 6	Verify the correctness of the install by executing the following command: CHKPRDOPT 0UNV511.
Step 7	Optionally, rename the UNVUBR511 subsystem to UBROKER for consistent operation across future installations.

Naming Conventions

In Workload Automation 5 for IBM i, some library names and object names include a **vrm** (version, release, and modification) suffix, **511**. This **vrm** suffix will change, as appropriate, for subsequent releases.

Additionally, some library names can be changed. These pages use the default names for these libraries:

- **UNVPRD511** (product library)
- **UNVTMP511** (temporary library)
- **UNVSPL511** (spool file library)
- **UNVCMDREF** (command reference library)

You can change the **UNVPRD511**, **UNVTMP511**, and **UNVSPL511** library names at installation time.

You can change **UNVCMDREF** only via the Universal Command Server configuration file.

Detailed Information

The following pages provide detailed information for IBM i Installation:

- Installation Package
- Installation Requirements

- Distribution File
- Transferring to IBM i
- Installation Procedures
- Customization
- Object Inventory Lists
- Licensing

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

IBM i Installation - Installation Package

- Package Components
- Component Compatibility

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Package Components

The Workload Automation for IBM i package includes the following Workload Automation components:

- Universal Broker 5.1.1
- Universal Command (Manager and Server) 5.1.1
- Universal Control (Manager and Server) 5.1.1
- Universal Copy 5.1.1
- Universal Data Mover (Manager and Server) 5.1.1
- Universal Display Log File for AS/400 log files 5.1.1
- Universal Encrypt 5.1.1
- Universal Message Translator 5.1.1
- Universal Query 5.1.1
- Universal Submit Job with z/OS WTO support 5.1.1

Workload Automation for IBM i is packaged with product identifier **0UNV511**. The product can be managed using the IBM i commands for working with licensed programs (such as **RSTLICPGM**, **SAVLICPGM**, and **DLTLICPGM**).

Note

For the 5.1.1 release, Universal Command and Universal Encrypt are packaged as one IBM i licensed program.

Component Compatibility

The following table identifies the compatibility of Workload Automation for IBM i with previous component / product versions.

Component	Compatibility
Universal Broker 5.1.1	Stonebranch Solutions / Universal Products 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Command 5.1.1	Universal Command 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Control 5.1.1	Universal Control 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Copy 5.1.1	Universal Copy 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Data Mover 5.1.1	Universal Data Mover 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Display Log File for IBM i log files 5.1.1	Universal Display Log File for OS/400 log files 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Encrypt 5.1.1	Universal Encrypt 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Message Translator 5.1.1	Universal Message Translator 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Query 5.1.1	Universal Broker 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Submit Job with z/OS WTO support 5.1.1	Universal Submit Job with z/OS WTO support 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.

The component references pertain to all support platforms for that version.

IBM i Installation - Installation Requirements

- System Requirements
- Platform Requirements
- Libraries and Logs
  - Universal Broker Job Log
  - Command Reference Library
  - Trace File Location Library
  - Spool Library

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

System Requirements

The requirements for installation of Workload Automation 5 for IBM i are:

- IBM i V5R4M0 or above.
- TCP/IP.
- User profile with *ALLOBJ, *SPLCTL, *JOBCTL, and *SECADM authorities.
- About 360 megabytes of disk space.
- Network-attached workstation.
- PTF SI27629 installed on V5R4 systems.
  For additional information, see the documentation for APAR SE28859 and PTF SI27629 at both:
  - https://www\-912.ibm.com/n_dir/nas4apar.nsf/aaf5d88f9cc2ee10862571020058635c/a2b5a3b3ac874a9d862572d8003c7457?OpenDocument&Highlight=2,QGYOLJBL
  - https://www\-912.ibm.com/a_dir/as4ptf.nsf/c2fd98f5d2eccb83862574ce00520341/7529dd654f63941b862572eb00581099?OpenDocument

The Workload Automation 5 for IBM i installation process creates a product user profile, **UNVUBR511**, that is given ***ALLOBJ** special authority. **UNVUBR511**, with ***ALLOBJ** special authority, is required to successfully complete the installation.

Note

Some organizations and companies require the removal of ***ALLOBJ** authority from non-administrative user profiles. Workload Automation may be configured to run without ***ALLOBJ** authority; however, to do so requires additional administrative overhead. The steps required to remove product ***ALLOBJ** authority are described in Universal Broker Security.

The ***BASE** option contains the complete installation. This includes all of the components and utilities listed above.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Libraries and Logs

Under the IBM i native file system, Workload Automation write to product files residing in various libraries and write information to job logs.

This section specifies the following information for various libraries and logs:

- Estimated amount of space required
- Required security access
- Location of the production files

Universal Broker Job Log

Under IBM i, Universal Broker writes its messages to the **UBROKER** job log.

Space

Job log file growth is dependent on use of the Workload Automation Servers. The disposition and size of the job log depends on the job definition as well as system variables **QJOBMSGQMX** (maximum job log size) and **QJOBMSGQFL** (action when job log is full).

Security

Since Universal Broker uses a normal job log, no special security is required.

Command Reference Library

Universal Command Server can execute commands of type **cmdref**. A command reference is a predefined command or script to which the Universal Command Manager refers by its file name.

The default command reference library name is **UNVCMDREF**. For security reasons, the name of this library can be changed only via the Universal Command Server configuration, located in the **UNVCONF** file and **UCMDS** member. The configuration file entry has precedence over the default value. If the name is changed to something other than the default or configured value, Universal Command Manager will no longer be able to read the command reference files; it will generate appropriate error messages if an attempt is made to do so.

Space

The amount of space is dependent solely on the number of command reference files defined.

Security

Universal Command Server requires read access to the **UNVCMDREF** library files. Administrator accounts require appropriate access in order to maintain the command reference files. No general user access is required.

Trace File Location Library

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. Customer Support to resolve product problems.

On IBM i, Workload Automation trace files normally are written to ***CURLIB** (current library). Under Universal Broker and the associated servers, the current library is the temporary library designated at installation time (**UNVTMP511**).

Otherwise, current library is the default current library of the user.

Space

Trace files can grow to significant size, depending on how long the trace is active and how much work the program is doing during the tracing period. Due to the information associated with IBM i pointers and fact that physical database files are fixed record lengths, trace files quickly can grow very large on a system with high Universal Broker and server activity.

If the trace file size is increased beyond 500,000 records, the maximum file size must be changed or the associated job will hang. The hang results from a system generated inquiry message, which is issued when the maximum file size is exceeded. By default, trace files wrap before reaching the maximum file size, thus avoiding the system inquiry message.

The MAX_TRACE_LINES configuration option sets the number of records at which the trace file wrapping occurs. When the maximum size is reached, the trace file will wrap to the beginning.

No space is required under normal operation for trace files. Trace files are requested by Stonebranch, Inc. Customer Support only for problem resolution. When trace files are required, at least 40MB of disk space should be available.

Security

The **UNVUBR511** user profile requires at least ***CHANGE** authority to the **UNVTMP511** library to create and use the Universal Broker and server trace files. No general user access is required.

Spool Library

The spool library is used to store the following types of information:

- Execution information for Workload Automation components started by Universal Broker, **UBR_CMP_DB**.
- Universal Command Server status, **SRV_CMP_DB**.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with manager fault tolerance enabled.

Spool files are stored in the **UNVSPL511** library.

Space

The spool files are located in the **UNVSPL511** library. The amount of disk space required for the spool directory depends on these factors:

1. Number of spooling user processes that will be executing simultaneously.
   A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50. Once a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
2. Average number of records of the user processes for standard input, output, and error files.
   Each record is 32764 bytes in length. Keep in mind that spooling is not intended as a feature for file transfer purposes. File transfer related processes should execute without spooling enabled.

Once these numbers are determined, the average amount of disk space is calculated with the following formula:

MAX-PROCESSES x AVERAGE NUMBER-OF-REC0RDS x 32764 = required disk space.

As an example, if the maximum number simultaneous user processes is estimated at 20 and the average number of records in files is 1,000 bytes, the average amount of required disk space is 655MB (20 x 1000 x 32764).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. The feature must be turned on through the ALLOW_SPOOLING configuration settings.

Further details on the Manager Fault Tolerant feature and the spooling of redirected standard I/O files can be found in the Universal Agent 7.2.x User Guide.

Security

The **UNVUBR511** user profile requires at least ***CHANGE** authority to the **UNVSPL511** library to create and use the spool files. No general user access is required.

No other Workload Automation 5 for IBM i components access the spool library.

IBM i Installation - Distribution File

- IBM i Distribution File
- Obtaining the Distribution File
- Distribution File Format
  - Full Release
  - Maintenance Release
- Distribution File Contents
  - Full Release
  - Maintenance Release

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

IBM i Distribution File

The Workload Automation for IBM i product distribution file contains all of the files required for the installation of the Workload Automation for IBM i package.

Platform	Distribution File
IBM i (AS/400 5.4)	**sb-5.1.1.0-as400-5.4.tar.Z**
IBM i 7.2	**sb-5.1.1.0-as400-7.2.tar.Z**

Obtaining the Distribution File

To obtain the Workload Automation for IBM i package, you must download the corresponding product distribution file from the Stonebranch Customer Portal.

Note

A customer user name and password — provided by Stonebranch, Inc. — are required to access this area.

Distribution File Format

The format of the Workload Automation for IBM i distribution file name is different for a full release than it is for a maintenance release.

Full Release

For a full release installation, the product distribution file name has the following format:

sb-Version.Release.Modification Level.Maintenance Level-operating system-version.release.tar.Z

For example: **sb-5.1.1.0-as400-5.4.tar.Z**

In this format:

- **Version** is the current version of Workload Automation.
- **Release** is the current release of Workload Automation.
- **Modification Level** is the current Workload Automation feature set.
- **Maintenance Level** is the Workload Automation build level.
- **operating system** is the name of the operating system.
- **version.release** is the minimum supported version and release of the operating system.

Maintenance Release

For a maintenance release installation, the product distribution file name has a modified format:

- **-ptfs** is included after the **operating system** name.
- **version.release** is not included after the **operating system** name.

For example: **sb-5.1.1.1-as400-ptfs.tar.Z**

(See Transferring for a Maintenance Release for information about PTFs in maintenance release distribution.)

Distribution File Contents

The distribution file contains the installation files required for the installation of Workload Automation for IBM i.

The Workload Automation Licensed Program for IBM i is distributed as an IBM i Save File.

Depending on whether the installation is for a full release or a maintenance release of Workload Automation for IBM i, the distribution file contains different installation files.

Full Release

The following table lists the installation files included in the distribution file for a full release of Workload Automation for IBM i.

File Name	Description
UNV511.SAVF	Workload Automation Licensed Program Save File.
README.TXT	Summary of the installation procedure.

Maintenance Release

The following table lists the installation files included in the distribution file for a maintenance release of Workload Automation for IBM i.

File Name	Description
CUMUNV511	Workload Automation Cumulative PTF Save File.
README.TXT	Summary of the installation procedure.

IBM i Installation - Transferring to IBM i

- Workload Automation for IBM i - Transferring Workload Automation to IBM i
- Transferring for a Full Release
- Transferring for a Maintenance Release

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Workload Automation for IBM i - Transferring Workload Automation to IBM i

In order to install Workload Automation for IBM i, you must first transfer the Workload Automation Licensed Program for IBM i to the IBM i operating system.

The transfer is comprised of three procedures:

1. Download the product distribution file from the Stonebranch Customer Portal (see IBM i Installation - Distribution File) to Windows, UNIX, or IBM i shell with tar and zcat utilities.
2. Extract all files from the distribution file.
3. Transfer the IBM i Save File (the Workload Automation Licensed Program) to a library on an iSeries.

This page provides separate transferring procedures for:

- Transferring for a Full Release
- Transferring for a Maintenance Release

Transferring for a Full Release

To transfer a full release package to IBM i — whether for a new installation of a full release or an upgrade installation to a new release — perform the following steps:

Step 1	Download the distribution file to a computer running a supported version of Windows or UNIX.
Step 2	Extract the installation files from the distribution file: • To extract the installation files on UNIX, QShell, or PASE shell, run: `zcat *.Z | tar xvf -` Note The space and the **-** character after **xvf** must be included. • To extract the installation files on Windows, use a utility capable of extracting files from a **.Z** file. The files will be extracted to the specified location.
Step 3	Create a save file on the target IBM i system in library **QGPL**. On the IBM i command line, execute the following command: `CRTSAVF FILE(QGPL/UNV511)`

Step 4	If needed, FTP the SAVF file extracted in Step 2 to the save file created in Step 3.
	
	Note
	The following example assumes that the SAVF file resides in: **`c:\temp\Workload Automation for as400`**
	
	`ftp your_as400`
	`Name: your_name`
	`Password:  your_password`
	`ftp> cd QGPL`
	`ftp> bin`
	`ftp> lcd "c:\temp\Workload Automation for as400"`
	`ftp> put UNV511.SAVF UNV511`
	`ftp> quit`

Transferring for a Maintenance Release

To upgrade Workload Automation from a currently installed 5.1.1 release to 5.1.1 maintenance release, cumulative PTFs are used. These cumulative PTFs are distributed as IBM i Save Files and can be downloaded from the Stonebranch Customer Portal. PTFs should be installed as user **QSECOFR** or a user with equivalent authority.

To transfer a maintenance release package to IBM i, perform the following steps:

Step 1	Download the distribution file to a computer running a supported version of Windows or UNIX.
Step 2	Extract the distribution file: • To extract the ***.tar.Z** file on UNIX, run: `zcat *.Z | tar xvf -` Note The space and the **-** character after **xvf** must be included. • To extract the distribution file for Windows, use a utility capable of extracting files from a **.Z** file. The files will be extracted to the specified location.
Step 3	Create a save file on the target IBM i system in library **QGPL**. On the IBM i command line, execute the following command: `CRTSAVF FILE(QGPL/CUMUNV511)`

Step 4	FTP the **.SAVF** file extracted in Step 2 to the save file created in Step 3.
	For example:
	
	`ftp your_as400`
	`Name: your_name`
	`Password:  your_password`
	`ftp> cd QGPL`
	`ftp> bin`
	`ftp> lcd c:\temp`
	`ftp> put CUMUNV511 CUMUNV511`
	`ftp> quit`

IBM i Installation - Installation Procedures

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Installation Processes

There are six different processes for installing Workload Automation on an IBM i system:

1. New Installation (Default)
2. New Installation (Custom)
3. Re-Installation of Same Release
4. Upgrade Installation to New Release
5. Propagating New Release to Additional Systems
6. Upgrade Installation for Maintenance Release

Your installation environment, currently installed Workload Automation 5 for IBM i release (if any), and Workload Automation 5 for IBM i release to be installed determine the process to use.

There are two additional procedures related to installation:

- UCHGRLS (Change Release Tag) Program for changing Workload Automation for IBM i command names in the IBM i **QSYS** library.
- Product Removal procedures that are required or optional, depending on the installation process.

Note

If the QCCSID system value of IBM i is configured to a non-default CCSID, you may need to perform a customization step following the product install. For information on how to perform this step, including additional information to help you determine if and why you may need to perform the step, see National Language Customization.

Installs requiring the customization may see errors referring to UNVTMPL and/or UNVCONF files in the UBROKER job log.

For example:

```
*NONE      Diagnostic                  06/10/21  11:44:09.256725  UBROKER      UNVPRD511   *STMT
*EXT                    *N

                                    From module . . . . . . . . :
XPSMESG

                                    From procedure  . . . . . . :
WriteLogfile

                                    Statement . . . . . . . . . :
19

                                    Message . . . . :   UNV3327E (1000000001) Error reading template
definitions

                                      in 'UNVTMPL(UACL)': XCBufferConfigFileContents, 5, bad or invalid
value
```

IBM i Installation - New Installation (Default)

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

New Installation (Default)

The default process for a new installation of Workload Automation for IBM i installs to the following libraries:

- Product library (**UNVPRD511**)
- Temporary library (**UNVTMP511**)
- Spool file library (**UNVSPL511)**
- Command reference library (**UNVCMDREF**)

V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

Step 1	Execute the following command to install Workload Automation to the default libraries:
	`RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511)`
	When installing Workload Automation on a version of IBMi that is newer than the version identified in the distribution file name (for example, installing the 5.4-targeted file on an IBMi version greater than 5.4 or the 7.2-targeted file on an IBMi 7.3 or higher system), add the FRCOBJCVN parameter to force object conversion to the newer platform.
	`RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511) FRCOBJCVN(*YES *ALL)`

Step 2	Verify that the installation was correct by executing the following command: `CHKPRDOPT 0UNV511` The following message should display: **No errors detected by CHKPRDOPT.**
Step 3	Optionally, rename the UNVUBR511 subsystem to UBROKER for consistent operation across future installations.
Step 4	Optionally, use the UCHGRLS (Change Release Tag) Program to change the names of the Workload Automation for IBM i commands in the IBM i **QSYS** library.

IBM i Installation - New Installation (Custom)

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

New Installation (Custom)

The custom method for a new installation of Workload Automation for IBM i allows installation to libraries other than **UNVPRD511**, **UNVTMP511**, and **UNVSPL511**, as required by your environment.

V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

Step 1	Execute the following command to install Workload Automation to user-specified libraries:
	
	`RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511) LIB(prodLib tmpLib spoolLib)`
	
	Note
	Replace **prodLib** with the desired product library name, **tmpLib** with the desired temporary library name, and **spoolLib** with the desired spool library name.
	Forced object conversion
	When installing on an IBMi version that is greater than the version identified in the distribution file name, add the FRCOBJCVN parameter to the RSTLICPGM command for force object conversion to the newer OS version.
	
	`RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511) LIB(prodLib tmpLib spoolLib) FRCOBJCVN(*YES *ALL)`
	
	For example, this would be needed when using the 5.4-targeted distribution file to install on IBMi versions greater than 5.4 or when using the 7.2-targeted distribution file to install on IBMi 7.3 or higher.

Step 2	Verify that the installation was correct by executing the following command: `CHKPRDOPT 0UNV511` The following message should display: **No errors detected by CHKPRDOPT.**
Step 3	Optionally, rename the UNVUBR511 subsystem to UBROKER for consistent operation across future installations.
Step 4	Optionally, use the UCHGRLS (Change Release Tag) Program to change the names of the Workload Automation for IBM i commands in the IBM i **QSYS** library.

IBM i Installation - Re-Installation of Same Release

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Re-Installation Using Same Libraries

If you are reinstalling a Workload Automation for IBM i release (5.1.1 and later) into the same libraries, you do not have to remove the currently installed release.

Note

Re-installing to the same libraries will preserve configuration and other user-modified files, as well as objects created as part of the installation process. The re-installation process will replace programs, commands, etc.

Re-installation to the same libraries can be performed without deleting the current release.

Step 1	Back up all file objects in all of the Workload Automation libraries - product, temporary, spool, and command reference - prior to re-installation (in case errors occur during the re-installation process).
Step 2	End the Universal Broker subsystem, **UNVUBR511** (**UBROKER** if renamed).
Step 3	Run **RSTLICPGM** to install Workload Automation using the library names used during the initial 5.1.1 installation.
Step 4	After installation is complete, check the configuration and component files to ensure that the re-installation process preserved all changes previously made. Restore any files not correctly preserved from the backed up file objects.
Step 5	Restart **UNVUBR511**.
Step 6	Make sure that previous events needed by the customer are displayed via the I-Activity Monitor and that any disconnected jobs, as well as any appropriate MFT spool files, are correctly displayed using the Universal Query. If problems exist, restore the spool library from the backed up file objects.
Step 7	If **UCHGRLS** was previously run to remove 511 tags from the Workload Automation commands in **QSYS**, manually delete the newly created, 511-tagged Workload Automation commands in **QSYS**. A list of those commands can be found in QSYS Library. Note Do not remove the **UNVUBR511** user profile from **QSYS**.

Re-Installation Using Different Library Names

Before reinstalling a Workload Automation for IBM i release (5.1.1 and later) into different libraries, you must remove the currently installed release.

Step 1	Save copies of all configuration files and component definition files; otherwise, the configuration and component information will be lost.
Step 2	Remove the currently installed release (see Product Removal).
Step 3	Re-install the 5.1.1 release. See New Installation (Default) and New Installation (Custom) to determine which process is most appropriate for your environment.
Step 4	Merge the saved configuration into the newly installed configuration files manually. Conversely, the saved configuration files can be copied in place of the reinstalled configuration files.
Step 5	Optionally, use the UCHGRLS (Change Release Tag) Program to change the names of the Workload Automation for IBM i commands in the IBM i **QSYS** library.

IBM i Installation - Upgrade Installation to New Release

- Upgrade Procedures
  - Print Files Removal
  - Currently Installed Release
- Install (Default or Custom)
- Post-Install Tasks
- Pre-Production Tasks
  - Change Release Tags
  - Change Port Address

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Upgrade Procedures

To upgrade Workload Automation for IBM i from a currently installed pre-5.1.1 release (release 5.1.0 or earlier) to a 5.1.1 release, the following procedures are required:

1. Install (Default or Custom)
2. Post-Install Tasks
3. Pre-Production Tasks

V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

Print Files Removal

In releases 4.1.0 - 4.3.0, Job Log Copy Keep used the **USBMJPRTF** and **UCMSVRPRTF** print files.

Starting with release 5.1.0, these print files are no longer used and will be removed during the installation process. The spooled job logs will not be lost because they are spooled to the **QEZJOBLOG** output queue.

To print job logs from a previous release, use the **CHGSPLFA** command to associate the job logs with a different printer file.

Currently Installed Release

You are not required to delete a currently installed pre-5.1.1 release (5.1.0 or earlier) of Workload Automation for IBM i before installing a 5.1.1 release. You can do so either before or after installing the 5.1.1 release (see Product Removal).

However, if you want to install a 5.1.1 release to the previously used installation libraries of a pre-5.1.1 release, you must first uninstall that pre-5.1.1 release:

Step 1	Save configuration, component, NLS, and any other files from the product library (**UNVPRD*** / **UNIVERSAL**) that may contain information needed after the new installation.
Step 2	Save any files from the command reference library (**UNVCMDREF)** and temporary library (**UNVTMP*** / **UNVTMP**) that are needed, as these libraries will be deleted by the uninstall process.
Step 3	Follow the release-appropriate uninstall process (see Product Removal) and the New Release installation procedure (see Install (Default or Custom), below, to install the new release.
Step 4	After the new installation is complete, merge required configuration and any other saved information as well as any needed files and/or members from the previous release into the new release.

Install (Default or Custom)

The processes (default or custom) for the upgrade installation to a new Workload Automation release (5.1.1) are the same as the processes for a new installation.

See New Installation (Default) and New Installation (Custom) to determine which process is most appropriate for your environment.

Post-Install Tasks

After the installation upgrade of 5.1.1 (or previous release) to Workload Automation, perform the following steps:

Step 1	Copy any customized code page files from the **UNVNLS** file members from the previous release to the newly installed **UNVNLS** file. Note Do not simply copy the old **UNVNLS** file to replace the new **UNVNLS** file. The new files contain many updates.
Step 2	If needed, manually update the newly installed configuration file, **UNVCONF**, from the file of the previous release. The new file contains many new configuration options. At a minimum, the Workload Automation license entries must be entered into the configuration file members for the managers of each licensed product.
Step 3	If the original component file (**UNVCOMP**) was modified, update the newly installed **UNVCOMP** file for each of the modified members.
Step 4	If either or both of the exit programs (**UCMSJOBI** and **UCMSJOBT**) in the previous release had been customized, you likewise must customize the newly installed exit programs. The exit programs are located in the product library, **UNVPRD511**. Note You cannot replace **UCMSJOBI** for 5.1.0 with programs of the same name from previous releases without a possible conflict. If you must use older programs because their source is no longer available, rename them and call them from the new exit program. See **UCMSJOBI** in Universal Command Server for IBM i - Commands for additional information.

Pre-Production Tasks

Change Release Tags

When you are ready to take the new release to production, you can use the **UCHGRLS** (Change Release Tag) program to:

- Rename the Workload Automation for IBM i commands in **QSYS** to the untagged command names in **UNVPRD511**.
- Tag the pre-5.1.1 release command names in **QSYS** with the version / release / modification number of that release (for example, **320**).

See UCHGRLS (Change Release Tag) Program for information on using **UCHGRLS**.

Change Port Address

No two Workload Automation installed under the same IBM-i can have the same port number associated with the Universal Broker subsystem (**UBROKER**) . Change the default port address, 7887, to another port number for all but one of the **UBROKER** subsystems. Workload Automation components and utilities intended to use the altered port number must specify a port number in the commands or program calls. Whether or not a command or program call is used depends on the platform from which the request is sent.

Note
Do not replace a configuration file with one from a different release.

IBM i Installation - Propagating New Release to Additional Systems

- Propagating New Release to Additional Systems
- SAVLICPGM and RSTLICPGM Method

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Propagating New Release to Additional Systems

You may want to install Workload Automation on a pre-production system, implement needed changes such as customizing configuration files, and then propagate these changes to other systems.

One method to propagate the customized product is via the **SAVLICPGM** and **RSTLICPGM** commands (see SAVLICPGM and RSTLICPGM Method, below).

Warning

A number of Stonebranch objects are created during the installation process. These include, but are not limited to:

- Commands in QSYS
- UBROKER job queue and class
- Job definitions in the product library
- Universal Broker Subsystem
- Print files in the product library.

The default command reference library, **UNVCMDREF**, is created during installation unless a library of that name exists on the system. This process only recreates the objects previously identified. It will not propagate changes made to those objects.

V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

SAVLICPGM and RSTLICPGM Method

The following steps are recommended:

Step 1	Install Workload Automation (see New Installation (Default) and New Installation (Custom)).
Step 2	Download the latest product maintenance from the Stonebranch Customer Portal and temporarily apply the PTFs. Allow the system to automatically handle any permanent PTF application via the **APYPTF** command. See Upgrade Installation for Maintenance Release for detailed instructions on applying a maintenance release.
Step 3	Customize the Workload Automation installation as described in Customization.
Step 4	Verify that the 5.1.1 Universal Broker Subsystem, **UNVUBR511**, is not active.
Step 5	To create a save file in which to save the customized product, invoke: `CRTSAVF FILE(QGPL/UNV511CUST) TEXT('Customized installation file for SB Prod 5.1.1')`
Step 6	Invoke: `SAVLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511CUST)`

Step 7	If needed, restart the Universal Broker Subsystem: `STRSBS UNVPRD511/UNVUBR511`
Step 8	Via what ever mechanism you prefer, move the save file to each system on which you want to restore the product.
Step 9	Invoke: `RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511CUST)`

Note

Although this procedure describes use of a save file, any other available save and restore device can be used.

IBM i Installation - Upgrade Installation for Maintenance Release

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Upgrade Installation for Maintenance Release

To upgrade Workload Automation for IBM i from a currently installed release to a maintenance release, perform the following steps to install the cumulative PTFs:

Step 1	If it is running, end the Universal Broker subsystem (**UNVUBR511**) on the IBM i system. On the IBM i, enter the following command: `===> ENDSBS SBS(UNVUBR511)`
Step 2	Load the PTF. On the IBM i, enter the following command: `===> LODPTF LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/CUMUNV511)`

Step 3	Apply the PTF.
	On the IBM i, enter the following command:
	`===> APYPTF LICPGM(0UNV511)`
	This command applies all PTFs contained in the cumulative PTF file.
	If you want to apply only selected PTFs, enter the following command:
	`===>APYPTF LICPGM(0UNV511) SELECT(ptf1 ptf2 ...)`
	This command applies the PTF temporarily.
	If you want to apply the PTFs permanently after successful testing, re-execute the command with the parameter **APY(*PERM)**.

Step 4	Restart **UNVUBR511**.

IBM i Installation - UCHGRLS (Change Release Tag) Program

- Overview
- Using UCHGRLS
- UCHGRLS Examples

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Overview

All installations of Workload Automation for IBM i, except an upgrade for maintenance, place the Workload Automation for IBM i commands in the product library (**UNVPRD511**) and the IBM i **QSYS** library.

Starting with the 3.2.0 release of Universal Products for IBM i, the command names in **QSYS** are tagged with the Workload Automation for IBM i version / release / modification number.

To maintain consistency across releases, you may prefer to use untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change command names in the IBM i **QSYS** library from tagged to untagged.

Using UCHGRLS

UCHGRLS resides in **UNVPRD511**. It provides two parameters, **SETTAGVRM** and **RMVTAGVRM**.

- **SETTAGVRM** specifies the release number of the release to be tagged.
- **RMVTAGVRM** specifies the release number of the tag to be removed.

Removing the tags changes the command names in **QSYS** to the normal production command names. (See UCHGRLS Examples, below.)

For a new installation, specify the following **UCHGRLS** command options and their values:

- Set tags (**SETTAGVRM**) option; value = ***NEW**.
- Remove tags (**RMVTAGVRM**) option; value = **511**.

This changes the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

For an upgrade installation (from a pre-5.1.1 release to 5.1.1), **UCHGRLS** tags the pre-5.1.1 release command names in **QSYS** with the version / release / modification number of that release (for example, **511**). See Object Inventory Lists for tagged command name details.

UCHGRLS Examples

This example runs **UCHGRLS** with the default values, following installation of **0UNV511** with **0UNV410** existing on the system.

```
UNVPRD511/UCHGRLS
```

This example runs **UCHGRLS** to revert back to **0UNV320** for the default names in **QSYS** with **0UNV511** installed. (This should be done prior to re-installing **0UNV511**, assuming that **0UNV320** remains installed on the system.)

```
UNVPRD511/UCHGRLS SETTAGVRM(511) RMVTAGVRM(320)
```

This example runs **UCHGRLS** to change the **0UNV511** commands in **QSYS** to the default names and to tag the **0UCM121** and **0UEN121** commands in **QSYS** with 121.

```
UNVPRD511/UCHGRLS SETTAGVRM(121) RMVTAGVRM(511)
```

This example runs **UCHGRLS** to set the **0UNV511** commands in **QSYS** to the default names, where **0UNV511** is the first Workload Automation installation on an IBM i system.

```
UNVPRD511/UCHGRLS SETTAGVRM(*NEW) RMVTAGVRM(511)
```

IBM i Installation - Product Removal

- Overview
- Universal Encrypt 1.2.1
- Universal Command 1.2.1
- Universal Products 3.1.1
- Universal Products 3.2.0
- Universal Products 4.1.0
- Stonebranch Solutions 4.2.0
- Stonebranch Solutions 4.3.0
- Workload Automation 5.1.0
- Workload Automation 5.1.1

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Overview

This page provides processes for the removal of current (5.1.1) and previous (5.1.0 or earlier) releases of Workload Automation for IBM i.

If you remove the currently installed release before installing or re-installing a 5.1.1 release, the removal process deletes the default command reference library, **UNVCMDREF**, in which the command reference files are located.

If you remove the currently installed release after installing or re-installing a 5.1.1 release, **UNVCMDREF** is not deleted.

To keep your command reference files from being deleted, rename **UNVCMDREF** and change the library owner. Release 5.1.1 will let you specify this library - or any library - as your command reference library via the CMD_REFERENCE_DIRECTORY option.

Before removing a 5.1.1 release, check CMD_REFERENCE_DIRECTORY in the Universal Command server member (**UCMDS**) of the Workload Automation configuration file (**UNVCONF**) - located in the product library (**UNVPRD511**); an alternate command reference library previously may have been selected. If it has, manually clear and delete the default command reference library, **UNVCMDREF**, after removing the 5.1.1 release.

Warning

If the **UCHGRLS** command was used to rename the 511-tagged names to normal production names (see UCHGRLS (Change Release Tag) Program), you must perform the following steps prior to uninstalling either 5.1.1, 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, or 1.2.1:

Step 1	Run the **UCHGRLS** command to temporarily rename the 5.1.1 Workload Automation commands in **QSYS** to the 511-tagged names and the tagged names of the older product to the normal production names. Without this step, Workload Automation may no longer run as expected.
Step 2	After removal of the older product is complete, use the ***NEW** option with the **UCHGRLS** command to change the 511-tagged names back to the normal, production names.

Universal Encrypt 1.2.1

Note

Before performing this uninstall, read the Overview.

You must uninstall Universal Encrypt 1.2.1 before uninstalling Universal Command 1.2.1.

The user ID used for uninstalling Universal Encrypt must have authority to delete all Universal Encrypt product-related objects.

To uninstall Universal Encrypt, perform the following steps:

Step 1	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UEN121)`
Step 2	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UEN121 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 1.

Universal Command 1.2.1

Note

Before performing this uninstall, read the Overview.

Before uninstalling Universal Command 1.2.1, you must uninstall Universal Encrypt 1.2.1.

The user ID used for uninstalling Universal Command must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Command, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UBROKER**). On the IBM i command line, execute the following command: `===>ENDSBS UBROKER`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UCM121)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UCM121 option *ALL release *ONLY not deleted."`** Correct the problem and repeat Step 2.

Universal Products 3.1.1

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Universal Products 3.1.1 must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Products 3.1.1, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UBROKER**). On the IBM i command line, execute the following command: `===>ENDSBS UBROKER`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV311)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV311 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

Universal Products 3.2.0

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Universal Products 3.2.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Products 3.2.0, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UNVUBR320**). On the IBM i command line, execute the following command: `===>ENDSBS UNVUBR320`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV320)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV320 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

Universal Products 4.1.0

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Universal Products 4.1.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Products 4.1.0, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UNVUBR410**). On the IBM i command line, execute the following command: `===>ENDSBS UNVUBR410`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV410)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV410 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

Stonebranch Solutions 4.2.0

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Stonebranch Solutions 4.2.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Stonebranch Solutions 4.2.0, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UNVUBR420**). On the IBM i command line, execute the following command: `===>ENDSBS UNVUBR420`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV420)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV511 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

Stonebranch Solutions 4.3.0

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Stonebranch Solutions 4.3.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Stonebranch Solutions 4.3.0, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UNVUBR430**). On the IBM i command line, execute the following command: `===>ENDSBS UNVUBR430`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV430)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV430 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

Workload Automation 5.1.0

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Workload Automation 5.1.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Workload Automation 5.1.0, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UNVUBR510**). On the IBM i command line, execute the following command: `===>ENDSBS UNVUBR510`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV510)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV511 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

Workload Automation 5.1.1

Note

Before performing this uninstall, read the Overview.

The user ID used for uninstalling Workload Automation 5.1.1 must have *ALLOBJ and *SECADM special authorities.

To uninstall Workload Automation 5.1.1, perform the following steps:

Step 1	If it is active, end the Universal Broker subsystem (**UNVUBR511**). On the IBM i command line, execute the following command: `===>ENDSBS UNVUBR511`
Step 2	To remove the licensed program, execute the following command: `===>DLTLICPGM LICPGM(0UNV511)`
Step 3	If the following message displays, use the **DSPJOBLOG** command to identify the problem: **`Objects for product 0UNV511 option *ALL release *ONLY not deleted.`** Correct the problem and repeat Step 2.

IBM i Installation - Customization

- Workload Automation for IBM i - Customization
- Modify the IPL Start-up Program
- Multiple-Installation Implementation
  - Universal Broker
  - Active Jobs
- Universal Broker Customization
  - Configuration
  - System Initialization
- Universal Command Manager Customization
  - Configuration
- Universal Command Server Customization
  - Configuration
- Universal Control Manager Customization
  - Configuration
- Universal Control Server Customization
  - Configuration
- Universal Data Mover Manager Customization
  - Configuration
- Universal Data Mover Server Customization
  - Configuration
- Universal Query Customization
  - Configuration
- National Language Customization

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Workload Automation for IBM i - Customization

(For information on applying product licenses to installed Workload Automation 5 for IBM i components, see IBM i Installation - Licensing.)

Modify the IPL Start-up Program

Modify the IPL Start-up Program so that the Universal Broker subsystem **(UNVUBR511)** is started at IPL time.

The subsystem start command is:

```
QSYS/STRSBS UNVPRD511/UNVUBR511
```

The IPL Start-up program name is displayed by:

```
DSPSYSVAL SYSVAL(QSTRUPPGM)
```

Refer to the AS/400 publication AS/400e series: Basic System Operation, Administration, and Problem Handling for complete details on modifying the IPL Start-up program.

Multiple-Installation Implementation

Workload Automation for IBM i provides the ability to install, configure, and test a new release prior to placing it into production.

You can keep an older release available during this initial production phase in case problems occur. This allows you to manually merge component files, configuration files, and exit programs without having to save copies of those files and programs prior to installation of the newer release.

When initially installed, the names of the Workload Automation commands in the **QSYS** library are tagged with the *v*ersion / *r*elease / *m*odification number, **511**. This renaming allows currently installed pre-5.1.1 releases to remain fully functional without modifying production code until pre-production testing is completed.

When you are ready to bring the newly installed release into production, run the UCHGRLS (Change Release Tag) Program to untag the 5.1.1 command names and tag the pre-5.1.1 command names with that pre-5.1.1 release number.

After running **UCHGRLS**, check the job log for errors because missing objects do not result in program termination. There will generally be one or more missing commands flagged; only those commands included with the installed release present a problem if they are missing.

Universal Broker

For the Universal Broker, both a job queue (***JOBQ**) and a class (***CLS**) were added; they are both named **UBROKER** and are located in the product library, **UNVPRD511**.

The **UNVUBR511** subsystem and Workload Automation job descriptions use the **UNVPRD511** / **UBROKER** class and job queue instead of the **QGPL** / **QBATCH** class and job queue.

To revert back to the **QGPL** / **QBATCH** class:

Step 1	Invoke the CHGRTGE command using SBSD(**UNVPRD511** / **UNVUBR511**), SEQNBR(**10**), and CLS(**QGPL** / **QBATCH**).
Step 2	Invoke the CHGPJE command using SBSD(**UNVPRD511** / **UNVUBR511**), PGM(**QSYS** / **QP0ZSPWP**), JOB(**UCMSINIT**), and CLS(**QGPL** / **QBATCH**).
Step 3	Invoke the CHGPJE command using SBSD(**UNVPRD511** / **UNVUBR511**), PGM(**QSYS** / **QP0ZSPWT**), JOB(**UNVSRV**), and CLS(**QGPL** / **QBATCH**).
Step 4	Use the RMVJOBQE command with SBSD(**UNVPRD511** / **UNVUBR511**) and JOBQ(**UNVPRD511** / **UBROKER**) to remove the job queue entry from the **UNVUBR511** subsystem.

To revert back to the **QGPL/QBATCH** job queue, invoke the CHGJOBD command specifying JOBQ(**QGPL** / **QBATCH**) for the **UNVPRD511** / **UBROKER**, **UNVPRD511** / **UCMSINIT**, and **UNVPRD511** / **UNVSRV** job descriptions.

Active Jobs

Whereas the **QGPL** / **QBATCH** job queue entry for the **QBASE** subsystem limits the maximum active jobs to only six (by default), the **UNVUBR511** / **UBROKER** job queue has no limit. Thus, it allows many additional Workload Automation jobs to run in parallel. Using the new settings, the Universal Broker RUNNING_MAX configuration option controls the maximum number of components allowed to run simultaneously.

For consistency with earlier releases, the initial settings of the **UNVUBR511** class is the same as the **QGPL** / **QBATCH** class. Using the new class and job description, you may better tune your systems without affecting those jobs running under the default **QGPL** / **QBATCH** class and job description.

Universal Broker Customization

Configuration

Configuration options for Universal Broker are stored in configuration file **UNVCONF**(**UBROKER**), in product library **UNVPRD511**.

See the Universal Broker 7.2.x Reference Guide for details on configuring Universal Broker.

The file can be edited using the Source Entry Utility (SEU).

System Initialization

Universal Broker runs as a job, UBROKER, in the Universal Broker subsystem (**UNVUBR511**) located in product library **UNVPRD511**.

To start Universal Broker, enter the following on the command line: **STRSBS UNVPRD511/UNVUBR511**. There are two sets of pre-start jobs initiated: **UNVSRV** and **UCMSINIT**:

- **UNVSRV** is the pre-start job for any of the Workload Automation servers.
- **UCMSINIT** is the pre-start job for the **UCMSINIT** program.

The Universal Command Server initiates **UCMSINIT**, which initiates and monitors the command originated from either the Universal Command Manager or the **exec** command under the Universal Data Mover Manager.

Universal Command Manager Customization

Configuration

Configuration options for Universal Command Manager are stored in its configuration file, member **UNVCONF**(**UCMD**) in product library **UNVPRD511**.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Manager.

The file can be edited using the Source Entry Utility (SEU).

Universal Command Server Customization

Configuration

Configuration options for Universal Command Server are stored in configuration file **UNVCONF**(UCMDS), in product library **UNVPRD511**.

See the Universal Command 7.2.x Reference Guide for details on configuring Universal Command Server.

Universal Command Server runs as a component managed by Universal Broker. It provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. The component definition file, **UNVCOMP**(**UCMD**), is located in product library **UNVPRD511**.

Universal Command Server uses the Universal Access Control List (UACL) configuration file as a level of product security. The UACL file is located in **UNVCONF**(**UACL**), in product library **UNVPRD511**. See the Universal Agent Security for information on how Universal Command Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

Universal Control Manager Customization

Configuration

Configuration options for Universal Control Manager are stored in configuration file **UNVCONF**(**UCTL**), in product library **UNVPRD511**.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Manager.

The file can be edited using the Source Entry Utility (SEU).

Universal Control Server Customization

Configuration

Configuration options for Universal Control Server are stored in configuration file **UNVCONF**(**UCTLS**), in product library **UNVPRD511**.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Control Server.

Universal Control Server runs as a component managed by Universal Broker. It provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. Component definition file **UNVCOMP**(**UCTL**) is located in library **UNVPRD511** by default

Universal Control Server uses the Universal Access Control List (UACL) configuration file as a level of product security. The UACL file **UNVCONF**(**UACL**) is located in product library **UNVPRD511**. See the Universal Agent Security for information on how Universal Control Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

Universal Data Mover Manager Customization

Configuration

Configuration options for Universal Data Mover Manager are stored in its configuration file, member **UNVCONF**(**UDM**) in product library **UNVPRD511**.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring Universal Data Mover Manager.

The file can be edited using the Source Entry Utility (SEU).

Universal Data Mover Server Customization

Configuration

Configuration options for Universal Data Mover Server are stored in configuration file **UNVCONF**(**UDMS**), in product library **UNVPRD511**.

See the Universal Data Mover 7.2.x Reference Guide for details on configuring Universal Data Mover Server.

Universal Data Mover Server runs as a component managed by Universal Broker. It provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. Component definition file **UNVCOMP(UDM)** is located in product library **UNVPRD511**.

Universal Data Mover Server uses the Universal Access Control List (UACL) configuration file as a level of product security. The UACL file is located in **UNVCONF**(**UACL**) in product library **UNVPRD511**. See the Universal Agent Security for information on how Universal Data Mover Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

Universal Query Customization

Configuration

Configuration options for Universal Query are stored in configuration file **UNVCONF**(**UQRY**), in product library*UNVPRD511*.

See the Universal Agent Utilities 7.2.x Reference Guide for details on configuring Universal Query.

The file can be edited using the Source Entry Utility (SEU).

National Language Customization

The default IBM i CCSID value (QCCSID) is pre-set to 65535 or *HEX, both of which mean: Do not interpret data unless specifically required. You can set QCCSID to a value consistent with your country CCSID.

However, Workload Automation configuration files contain non-invariant characters (such as #, which marks the start of a comment). Some CCSIDs, such as 278, result in the translation of these characters into characters that the Workload Automation software cannot recognize. As a result, Workload Automation components, including Universal Broker, fail to start.

To resolve this, either:

- Use a Universal Broker job CCSID of 65535. Both the Workload Automation configuration files and the internal Universal Broker code are CCSID 037-based. Start Universal Broker and check the job log output to verify the correct translation.
- Use 037 as the CCSID for the Universal Broker user profile (**UNVUBR511**). This sets the CCSID of the associated Universal Broker job to 037. The job's 37 CCSID forces compatibility with the configuration files and the internal program CCSID value of 037, which is consistent with standard IBM i development practice.

The **CRTPGM** command sets the program CCSID to 65535, which allows the program to inherit the job CCSID. Writing to the job log should result in the correct translation of characters between the job CCSID and the system CCSID.

IBM i Installation - Object Inventory Lists

- Workload Automation for IBM i Objects
- Installation Libraries
  - QSYS Library Command Names
- Source File Record Lengths
- Product Library (UNVPRD511)
- Temporary Library (UNVTMP511)
- Spool Library (UNVSPL511)
- Command Reference Library (UNVCMDREF)
- QSYS Library

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Workload Automation for IBM i Objects

The Workload Automation for IBM i installation includes the objects required for:

- Universal Broker
- Universal Command Manager and Server
- Universal Control Manager and Server
- Universal Data Mover Manager and Server
- Universal Encrypt

Installation Libraries

These objects are contained in four user-designated installation libraries:

- Product library **UNVPRD511** (formerly **UNIVERSAL**)
  Library containing all Workload Automation for IBM i products.
- Temporary library **UNVTMP511** (formerly **UNVTMP**)
  Library used as a temporary work area by Workload Automation. It may contain such items as trace files and temporary job log files.
- Spool library **UNVSPL511** (formerly **UNVSPOOL**)
  Library containing a spool database of all Workload Automation spool files. The database is allocated the first time that Universal Broker is started.
- Command reference library **UNVCMDREF**
  Library containing user-defined command references

In addition, an installation places the following in the IBM i **QSYS** library:

- Copy of the Workload Automation for IBM i commands contained in **UNVPRD511**.
- Universal Broker user profile, **UNVUBR511**.

QSYS Library Command Names

The names of the commands in **QSYS** are tagged with the Workload Automation for IBM i version / release / modification number, **511**. However, since command names can have a maximum nine characters, the following names are shortened:

- **UDSPLOGF** becomes **UDSPLF511**.
- **USBMJOB** becomes **USBMJ511**.
- **UMSGHNDLR** becomes **UMSGH511**.
- **UJOBINIT** becomes **UJOBI511**.

Note

For further information on library and object names, see Naming Conventions.

Source File Record Lengths

Workload Automation for IBM i source file record lengths must be a maximum 240 characters in order to use the Source Edit Utility, SEU.

The configuration file (**UNVCONF**), component file (**UNVCOMP**), template file (**UNVTMPL**), National Language Support file (**UNVNLS**), and the source files - all of which are located in the Workload Automation installation library (**UNVPRD511**, by default) - are editable by SEU as they are initially installed.

Product Library (UNVPRD511)

Name	Type	Description

PRCDSPLOGF	*PGM	Universal Display Log File program.
PUMSGHNDLR	*PGM	Universal Submit Job, WTO msg callback program.
STRUCP	*PGM	UCOPY command program.
STRUEN	*PGM	Universal Encrypt command program.
STRUME	*PGM	UMET command program.
UBROKER	*PGM	Universal Broker program.
UCMCPP	*PGM	UCMD Manager command processing program.
UCHGRLS	*PGM	Universal Change Release program.
UCMD	*PGM	UCMD Manager program.
UCMDEF01	*PGM	UCMD object definition program.
UCMDEXIT01	*PGM	UCMD install exit program.
UCMMSG01	*PGM	UCMD message definition program.
UCMSINIT	*PGM	UCMD Server Initiator program.
UCMSJOBI	*PGM	UCMD Server Initiator job initialization program.
UCMSJOBT	*PGM	UCMD Server Initiator job termination program.
UCMSRV	*PGM	Universal Command Server program.
UCOPY	*PGM	Universal Copy program.
UCTCPP	*PGM	UCTL Manager command processing program.
UCTL	*PGM	Universal Control Manager program.
UCTSRV	*PGM	Universal Command Server program.
UDM	*PGM	Universal Data Mover Manager program.
UDMCPP	*PGM	UDM Manager command processing program.
UDMSRV	*PGM	UDM Server program.
UDSPLOGF	*PGM	Universal Display Log File cmd processing program.
UENCRYPT	*PGM	Universal Encrypt program.
UJOBINIT	*PGM	USBMJOB initialization program.
ULSTSECPP	*PGM	Universal Spool List utility command processing program.
UMET	*PGM	Universal Message Translator program.
UMSGHNDLR	*PGM	Universal Submit Job, PUMSGHNDLR interface.
UQRCPP	*PGM	Universal Query command program.
UQUERY	*PGM	Universal Query program.
URMVSECPP	*PGM	Universal Spool Remove utility command processing program.
USBMJOB	*PGM	Universal Submit Job program.
USBMUSRJOB	*PGM	Universal Submit User Job program.
USLIST	*PGM	Universal Spool List Job program.
USLRM	*PGM	Universal Spool Remove Job program.
UCMINST	*MSGF	UCMD install message file.
UNVMSG	*MSGF	Workload Automation message file.
UNVSEQMSGF	*MSGF	Universal Products Sequential Message File.
USBMJOB	*MSGF	Universal Submit Job message file.
CP2CCSID_X	*FILE	Example code page to CCSID mapping file.

INSTL.INFO	*FILE	Installation Information file for library locations.
UCMNLSTMP	*FILE	Temporary National Language Support file.
UCMSVRPRTF	*FILE	Universal Command server job log print file.
UNVCLSRC	*FILE	CL source.
UNVCOMP	*FILE	Universal component definition members.
UNVCONF	*FILE	Universal configuration members.
UNVDDSSRC	*FILE	DDS Src for Stonebranch Products.
UNVNLS	*FILE	National Language Support.
UNVTMPL	*FILE	Universal template members.
USBMJPRTF	*FILE	USBMJOB job log print file.
UBROKER	*JOBD	Job Definition for Universal Broker.
UCMSINIT	*JOBD	Job Definition for UCMD Server Initiator.
UNVSRV	*JOBD	Job Definition for Universal Servers.
UBROKER	*JOBQ	Universal Broker job queue.
UBROKER	*CLS	Universal Broker class.
STRUCM	*CMD	Universal Command Manager command.
STRUCP	*CMD	Universal Copy command.
STRUCT	*CMD	Universal Control Manager command.
STRUDM	*CMD	Universal Data Mover Manager command.
STRUEN	*CMD	Universal Encrypt command.
STRUME	*CMD	Universal Message Translator command.
STRUQR	*CMD	Universal Query command.
UCHGRLS	*CMD	Universal Change Release command.
UDSPLOGF	*CMD	Universal Display Log File command.
UJOBINIT	*CMD	USBMJOB initialization program command.
ULSTSE	*CMD	Universal Spool List utility command.
UMSGHNDLR	*CMD	WTO message callback program command.
URMVSE	*CMD	Universal Spool Remove utility command.
USBMJOB	*CMD	Universal Submit Job command.
UNVUBR511	*SBSD	Subsystem Definition for Universal Broker
STRUCM	*PNLGRP	UCMD Manager panel definition.
STRUCP	*PNLGRP	UCOPY panel definition.
STRUCT	*PNLGRP	UCTL Manager panel definition.
STRUDM	*PNLGRP	UDM Manager panel definition.
STRUEN	*PNLGRP	Universal Encrypt panel definition.
STRUME	*PNLGRP	UMET panel definition.
STRUQR	*PNLGRP	Universal Query panel definition.
UCHGRLS	*PNLGRP	Universal Change Release panel definition.
UDSPLOGF	*PNLGRP	Universal Display Log File panel definition.
UJOBINIT	*PNLGRP	USBMJOB initialization program panel definition.
ULSTSE	*PNLGRP	Universal Spool List utility panel definition.

UMSGHNDLR	*PNLGRP	WTO message callback program panel definition.
URMVSE	*PNLGRP	Universal Spool Remove utility panel definition.
USBMJOB	*PNLGRP	Universal Submit Job panel definition.
UNV511	*PRDDFN	Workload Automation product definition.
UNV511	*PRDLOD	Workload Automation product load.

Temporary Library (UNVTMP511)

Name	Type	Description
UCMDEF02	*PGM	UNVTMP object definition program.
UCMDEXIT02	*PGM	UNVTMP install exit program.

Spool Library (UNVSPL511)

Name	Type	Description
UNVDEFSPL	*PGM	**UNVSPL511** object definition program.
UNVEXITSPL	*PGM	**UNVSPL511** install exit program.

Command Reference Library (UNVCMDREF)

Name	Type	Description
UNVCRFSRC	*FILE	Universal Command command reference file.

QSYS Library

Name	Type	Description
STRUCM511	*CMD	Universal Command Manager command.
STRUCP511	*CMD	Universal Copy command.
STRUCT511	*CMD	Universal Control Manager command.
STRUDM511	*CMD	Universal Data Mover Manager command.
STRUEN511	*CMD	Universal Encrypt command.
STRUME511	*CMD	Universal Message Translator command.
STRUQR511	*CMD	Universal Query command.
UDSPLF511	*CMD	Universal Display Log File command.
UJOBI511	*CMD	Universal Job Initializer command.
ULSTSE511	*CMD	Universal Spool List utility command.
UMSGH511	*CMD	Universal Message Handler command.
URMVSE511	*CMD	Universal Spool Remove utility command.
USBMJ511	*CMD	Universal Submit Job command.
UNVUBR511	*USRPRF	Universal Broker user profile.

IBM i Installation - Licensing

- Licensing Workload Automation 5 for IBM i Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Licensing Workload Automation 5 for IBM i Components

After Workload Automation 5 for IBM i has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager

Product License File

Product license information (license parameter keywords and their values) is contained in a text file provided by your Stonebranch, Inc. account representative.

Format

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: **Indesca_Stonebranch_OS400_A1.txt**.

- For Universal Command Manager, **Indesca** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.
- For Universal Data Mover Manager, **Infitran** is used as the <*component name*> in the product license file name and as the name of the product in the file itself.

Sample

The following is a sample Universal Command Manager for IBM i product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "OS400"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Entering License Information

In the **UNVPRD511** product library:

- Enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UNVCONF**(**UCMD**).
- Enter the Universal Data Mover Manager license parameters into the Universal Data Mover Manager configuration file, member **UNVCONF**(**UDM**).

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it:

Stop Universal Broker	`ENDSBS UBROKER *CNTRLD` (`*CNTRLD` performs a controlled shutdown.)
Start Universal Broker	`STRSBS UBROKER`

Verifying Universal Agent Installation

Verifying Agent Installation

If an installed Agent is configured to communicate with Universal Controller, you can verify that it is installed, running, and communicating with the Controller by performing the following steps:

Step 1

From your browser, open the Universal Controller user interface (where `localhost` is the name of the machine):

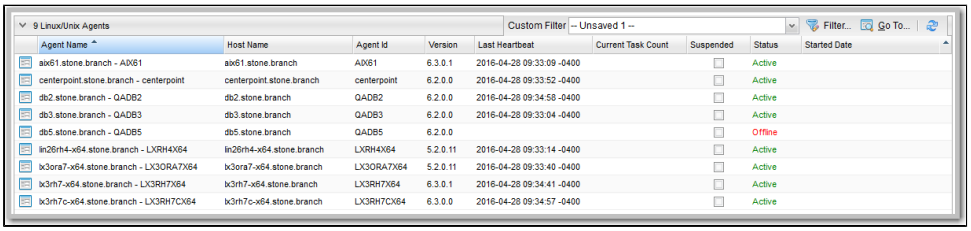
```
http://localhost:8080/opswise
```

Step 2

Log in with a user name and password for this system.

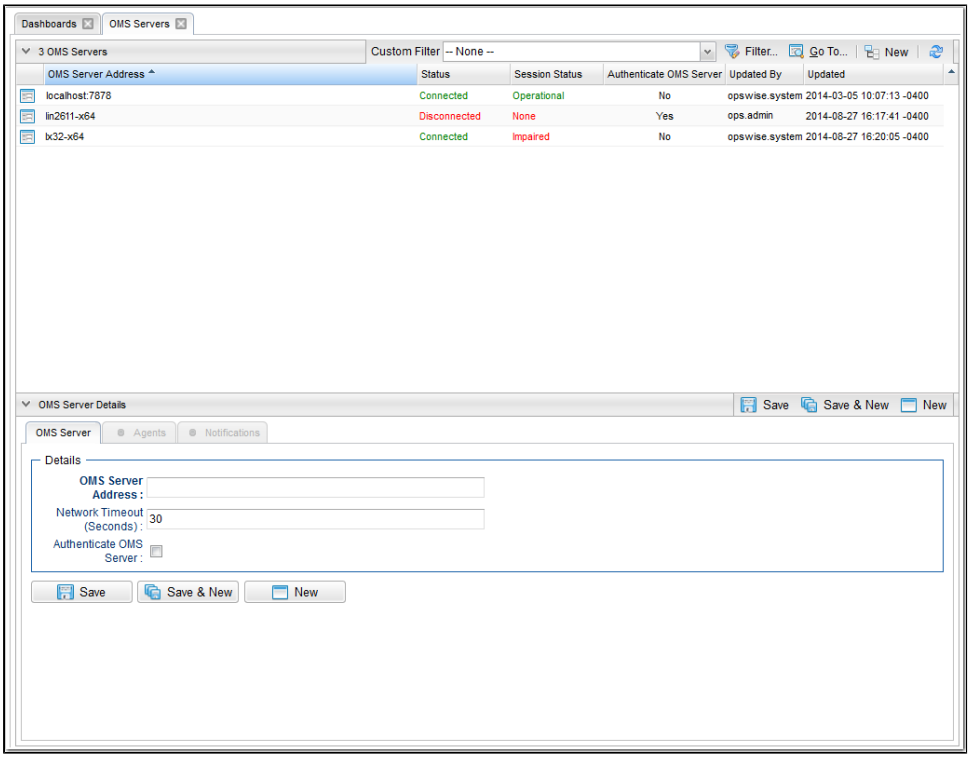
Step 3

To check for your newly installed Agent, select **Agents > All Agents** from the Agents & Connections navigation pane. You will see a list similar to the following example, which includes the Agent that you just installed. Make sure the Agent Status is **Connected**.



Step 4

To check for your newly installed OMS Server, select **System > OMS Servers** from the Agents & Connections navigation pane. You will see a list similar to the following example, which includes the OMS Server that you just installed as part of the Agent package. Make sure the OMS Server Status is **Active**.



Step 5

For more information about these components in the Universal Controller user interface, see:

- Agents
- OMS Servers

Upgrading Universal Agent

- Overview
- Licensing
- Upgrade Procedures
  - Upgrading Universal Agent for z/OS
  - Upgrading Universal Agent for Windows
  - Upgrading Universal Agent for SOA for Windows
  - Upgrading Universal Enterprise Controller (UEC) for Windows
  - Upgrading UEC Client Applications for Windows
  - Upgrading Universal Agent for UNIX
  - Upgrading Universal Agent for SOA for UNIX
  - Upgrading Universal Agent for IBM i

Overview

Upgrading Universal Agent refers to the increase of a currently installed pre-7.2.x Version, Release, or Modification level (see Versioning) of Universal Agent (7.1.x, 7.0.x, 6.9.x, 6.8.x, 6.7.x, 6.6.x, 6.5.x, 6.4.x, 6.3.x, 6.2.x, 5.2.0, 5.1.0, 4.x, 3.x, 2.x) to Universal Agent 7.2.x.

Universal Agent refers to all Universal components that have been packaged under the following names:

- Universal Agent 7.2.x
- Universal Agent 7.1.x
- Universal Agent 7.0.x
- Universal Agent 6.9.x
- Universal Agent 6.8.x
- Universal Agent 6.7.x
- Universal Agent 6.6.x
- Universal Agent 6.5.x
- Universal Agent 6.4.x
- Universal Agent 6.3.x
- Universal Agent 6.2.x
- Opswise Universal Agent 5.2.0
- Workload Automation 5.1.0 (Indesca, Infitran, UAG)
- Indesca / Infitran 4.x
- Universal Products 3.x
- Universal Command 2.x

You do not have to remove a pre-7.2.x Agent to upgrade to Agent 7.2.x.

(Applying maintenance to an Agent refers to the increase of its currently installed Maintenance level to a later Maintenance level; for example, applying maintenance to Agent 7.2.0.0 to increase its maintenance level to Agent 7.2.0.1.)

Note

You cannot upgrade an Opswise Automation Center Agent (releases 1.7, 1.6, and 1.5) to Universal Agent 7.2.x.

IMPORTANT

Before upgrading Universal Connector, you must first install the appropriate SAP NW RFC Libraries for the platform on which the connector agent is installed (see Installing Universal Agent - Universal Connector for SAP Prerequisite).

Licensing

All upgrades to Universal Agent 7.2.x from versions 4.x, 5.1.0, 5.2.0, 6.2.x, 6.3.x, 6.4.x, 6.5.x, 6.6.x, 6.7.x, 6.8.x, 6.9.x, 7.0.x, and 7.1.x do not require new licenses for its licensed components, except for the UEC, UEM, and SOA licensed components of version 4.1.0. That SOA component will require a new license. Licenses are no longer required for UEC and UEM components.

All upgrades to Universal Agent 7.2.x from versions 3.2.0 and earlier do require new licenses for its licensed components.

Upgrade Procedures

The upgrade procedure actually is an un-install / re-install of the Agent. All configuration files (*.conf) are updated, not replaced. Local modifications to the configuration files are preserved.

Upgrading Universal Agent for z/OS

Step 1	From the Stonebranch Customer Portal, download the Agent product distribution file for z/OS (for instructions, see Product Distribution File Download).
Step 2	Stop all started tasks (STC) on the Agent that you are updating (for instructions on how to stop an Agent on a specific platform, see Starting and Stopping Agent Components).
Step 3	Install the Agent. For version-specific details, see z/OS Installation - SMP/E Installation.
Step 4	Start the Agent. The start/stop procedure may differ depending on your platform (see Starting and Stopping Agent Components and select your platform).
Step 5	Verify that the Agent is installed and running. If the Agent is configured to communicate with a Controller, verify that the Controller shows the Agent as being online (see Verifying Universal Agent Installation).

Upgrading Universal Agent for Windows

Step 1	From the Stonebranch Customer Portal, download the Agent product distribution file that is appropriate for your platform (for instructions, see Product Distribution File Download).
Step 2	Install the Agent. For details, see Windows Installation.
Step 3	Verify that the Agent is installed and running. If the Agent is configured to communicate with a Controller, verify that the Controller shows the Agent as being online (see Verifying Universal Agent Installation).

Note

A user mode installation of Universal Agent for Windows does not support an upgrade (that is, configuration files are not preserved.)

Configuration File Archives

During a Windows system mode install upgrade, the installation logic creates .archive directories (if they do not already exist) under the `C:\ProgramData\Universal\conf` and the `C:\ProgramData\Universal\comp` directories. Copies of existing configuration and component definition files, respectively, are stored in these directories. These archived files are then used to merge existing configuration and component definition values into the newly installed files. This allows the install to provide new configured default values while retaining existing settings.

If you take your own configuration or component definition file backups prior to upgrading a Universal Agent install, be sure to save them outside of the `.\conf\.archive` and `.\comp\.archive` directories. Otherwise, the configuration merge logic may fail to behave as expected.

Upgrading Universal Agent for SOA for Windows

Step 1	From the Stonebranch Customer Portal, download the Universal Agent for SOA for Windows product distribution file (for instructions, see Product Distribution File Download).
Step 2	Perform the recommended back-up of Universal Agent for SOA files (see Universal Agent for SOA for Windows - Pre-Installation / Upgrade Backups).
Step 3	Install Universal Agent for SOA. For details, see Universal Agent for SOA for Windows Installation.

Upgrading Universal Enterprise Controller (UEC) for Windows

Step 1	From the Stonebranch Customer Portal, download the Universal Enterprise Controller (UEC) for Windows product distribution file (for instructions, see Product Distribution File Download).
Step 2	Install UEC. For details, see Universal Enterprise Controller for Windows Installation.

Upgrading UEC Client Applications for Windows

Step 1	From the Stonebranch Customer Portal, download the UEC Client Applications for Windows product distribution file (for instructions, see Product Distribution File Download).
Step 2	Log out of active UEC Client Applications instances on the machine where you are upgrading UEC Client Applications.
Step 3	Install UEC Client Applications. For details, see UEC Client Applications Installation.

Note

If you are upgrading a per-user installation of UEC Client Applications from any release prior to 5.2.0.1, you must uninstall the older version before installing the new version. There is an error in the older releases that prevents the upgrade from retaining a per-user installation environment.

Upgrades from 5.2.0.1 to any newer release should behave as expected.

Upgrading Universal Agent for UNIX

Step 1	From the Stonebranch Customer Portal, download the Agent product distribution file that is appropriate for your platform (for instructions, see Product Distribution File Download).
Step 2	Install the Agent. For details, see Universal Agent for UNIX Installation.
Step 3	If you are installing the Agent on AIX or Linux, you must review and perform PAM customization: • AIX PAM Customization • Linux PAM Customization
Step 4	Verify that the Agent is installed and running. If the Agent is configured to communicate with a Controller, verify that the Controller shows the Agent as being online (see Verifying Universal Agent Installation).

Note

A user mode installation of Universal Agent for UNIX does not support an upgrade (that is, configuration files are not preserved.)

Configuration File Archives

During a UNIX system mode install upgrade, the installation logic creates .archive directories (if they do not already exist) under the `/etc/universal` and `/etc/universal/comp` directories. Copies of existing configuration and component definition files, respectively, are stored in these directories. These archived files are then used to merge existing configuration and component definition values into the newly installed files. This allows the install to provide new configured default values while retaining existing settings.

If you take your own configuration or component definition file backups prior to upgrading a Universal Agent install, be sure to save them outside of the `/etc/universal and /etc/universal/comp` directories. Otherwise, the configuration merge logic may fail to behave as expected.

Upgrading Universal Agent for SOA for UNIX

Step 1	From the Stonebranch Customer Portal, download the Universal Agent for SOA for UNIX product distribution file that is appropriate for your platform (for instructions, see Product Distribution File Download).
Step 2	Perform the recommended back-up of Universal Agent for SOA files (see Universal Agent for SOA for UNIX - Pre-Installation / Upgrade Backups).
Step 3	Install the Universal Agent for SOA. For details, see Universal Agent for SOA for UNIX Installation.

Upgrading Universal Agent for IBM i

Currently, IBM i runs Workload Automation 5.1.1. For information on upgrading to that version, see IBM i Installation - Upgrade Installation to New Release.

Applying Maintenance to Universal Agent

- Overview
- Applying Maintenance to Linux/Unix and Windows Agents
- Applying Maintenance to z/OS Agents

Overview

Applying maintenance to Universal Agent refers to the increase from its currently installed Maintenance level (see Versioning) to a later Maintenance level; for example, applying maintenance to a 7.2.0.0 Agent to increase its maintenance level to 7.2.0.1.

(Upgrading an Agent refers to the increase from its currently installed Version, Release, or Modification level to a later Version, Release, or Modification level; for example, upgrading a 6.2.0.1 Agent to a 7.2.0.0 Agent.)

Applying Maintenance to Linux/Unix and Windows Agents

The procedure for applying maintenance to Linux/Unix and Windows Agents is the same as the procedure for upgrading Linux/Unix and Windows Agents, which includes downloading and installing a platform-specific Agent from the Stonebranch Customer Portal.

Applying Maintenance to z/OS Agents

To apply maintenance to Universal Agent for z/OS:

Step 1	From the Stonebranch Customer Portal, download the **z/OS - PTFs package** file (for instructions, see Product Distribution File Download).
Step 2	Stop all Agent processes on the Agents to which you are applying maintenance (for instructions on how to stop the Agent on a specific platform, see Starting and Stopping Agent Components).
Step 3	Apply the maintenance to the Agent using the instructions in the README file that is included with **z/OS PTFs**.
Step 4	Verify that the Agent is communicating with Universal Controller (see Verifying Universal Agent Installation).

Licenses for Third-Party Libraries

- Overview
- Universal Agent
  - Berkeley DB License
  - OpenSSL License
  - zlib License
  - ICU License
  - Boost Software License
  - libcurl License
  - libssh2 License
  - Standard Function Library (SFL) License
  - Python Software Foundation License Version 2
- Universal Enterprise Controller
  - xerces License
- Universal Enterprise Controller Client Applications
  - prefuse License
- Universal Agent for SOA
  - Apache 2.0 License
    - Apache Axis2 Notice
    - Apache Tomcat Notice
    - Apache Derby Notice
    - Apache log4j Notice
  - Open Java Run Time Environment (Version 1.8)
    - OPENJDK ASSEMBLY EXCEPTION
    - The GNU General Public License (GPL)
    - ASM Bytecode Manipulation Framework v5.0.3
    - BSDiff v4.3
    - CodeViewer 1.0
    - Cryptix AES 3.2.0
    - CUP Parser Generator for Java 0.10k
    - DejaVu fonts v2.34
    - Document Object Model (DOM) Level 2 & 3
    - Dynalink v0.5
    - Elliptic Curve Cryptography
    - ECMAScript Language Specification ECMA-262 Edition 5.1
    - freebXML Registry 3.0 & 3.1
    - IAIK PKCS#11 Wrapper
    - ICU4C 4.0.1 and ICU4J 4.4
    - IJG JPEG 6b
    - Jing 20030619
    - Joni v1.1.9
    - JOpt-Simple v3.0
    - Kerberos functionality
    - Kerberos functionality from FundsXpress, INC.
    - Kronos OpenGL headers
    - Portions Copyright Eastman Kodak Company
    - GIFLIB 5.1.1 & libungif 4.1.3
    - Little CMS 2.9
    - Mesa 3D Graphics Library v4.1
    - Mozilla Network Security Services (NSS)
    - PC/SC Lite for Suse Linux v.1.1.1
    - PorterStemmer v4
    - Relax NG Object/Parser v.20050510
    - RelaxNGCC v1.12
    - Relax NG Datatype 1.0
    - SoftFloat version 2b
    - Sparkle 1.5
    - Thai Dictionary
    - Unicode 6.2.0 & CLDR 21.0.1
    - UPX v3.01
    - Xfree86-VidMode Extension 1.0
    - X Window System 6.8.2
    - zlib v1.2.11
    - Apache

Overview

This page provides the following license files for the third-party libraries used within Universal Agent:

Universal Agent

- Berkeley DB License

- OpenSSL License
- zlib License
- ICU License
- BOOST Software License
- libcurl License
- libssh2 License
- Standard Function Library (SFL) License
- Python Software Foundation License Version 2

Universal Enterprise Controller

- xerces License

Universal Enterprise Controller Client Applications

- prefuse License

Universal Agent for SOA

- Apache 2.0 License
  - Apache Axis2 Notice
  - Apache Tomcat Notice
  - Apache Derby Notice
  - Apache log4j Notice

Universal Agent

Berkeley DB License

The Berkeley DB library is used in binary form on z/OS, Windows, and UNIX ports.

The following is the license that applies to this copy of the Berkeley DB
software. For a license to use the Berkeley DB software under conditions
other than those described here, or to purchase support for this software,
please contact Oracle at berkeleydb-info_us@oracle.com.

==========
/*
* Copyright (c) 1990-2009 Oracle. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Redistributions in any form must be accompanied by information on
* how to obtain complete source code for the DB software and any
* accompanying software that uses the DB software. The source code
* must either be included in the distribution or be available for no
* more than the cost of distribution plus a nominal fee, and must be
* freely redistributable under reasonable conditions. For an
* executable file, complete source code means the source code for all
* modules it contains. It does not include source code for modules or
* files that typically accompany the major components of the operating
* system on which the executable file runs.
*
* THIS SOFTWARE IS PROVIDED BY ORACLE ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR
* NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL ORACLE BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
* BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
* OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
* IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/*
* Copyright (c) 1990, 1993, 1994, 1995
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without

```
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
 /*
 * Copyright (c) 1995, 1996
 * The President and Fellows of Harvard University. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
==========================
/***
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2005 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 * contributors may be used to endorse or promote products derived from
 * this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
```

```
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

OpenSSL License

The OpenSSL toolkit is used in binary form on z/OS, Windows, UNIX, and IBM i ports.

```
LICENSE ISSUES
==============

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License
---------------

/* ====================================================================
* Copyright (c) 1998-2017 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------
```

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the routines from the library
* being used are not cryptographic related

:-)

.
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

zlib License

The zlib library is used in binary form on z/OS, Windows, UNIX, and IBM i ports.

The following is the zlib library license.

(C) 1995-2004 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

ICU License

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2015 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.
Third-Party Software Licenses
This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.
1. Unicode Data Files and Software

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2015 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining
a copy of the Unicode data files and any associated documentation
(the "Data Files") or Unicode software and any associated documentation
(the "Software") to deal in the Data Files or Software
without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that
(a) this copyright and permission notice appear with all copies
of the Data Files or Software,
(b) this copyright and permission notice appear in associated
documentation, and
(c) there is clear notice in each modified Data File or in the Software
as well as in the documentation associated with the Data File(s) or
Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.

2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

1. The Google Chrome software developed by Google is licensed under the BSD license. Other software included in this distribution is provided under other licenses, as set forth below.
2. The BSD License
3. http://opensource.org/licenses/bsd-license.php
4. Copyright (C) 2006-2008, Google Inc.
5. All rights reserved.
6. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
7. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
8. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
9. Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
11. The word list in cjdict.txt are generated by combining three word lists listed
12. below with further processing for compound word breaking. The frequency is generated
13. with an iterative training against Google web corpora.
14. * Libtabe (Chinese)
15. - https://sourceforge.net/project/?group_id=1519
16. - Its license terms and conditions are shown below.
17. * IPADIC (Japanese)
18. - http://chasen.aist-nara.ac.jp/chasen/distribution.html
19. - Its license terms and conditions are shown below.
20. --------~~COPYING.libtabe ---- BEGIN~~----------
21. /*
22. * Copyrighy (c) 1999 TaBE Project.
23. * Copyright (c) 1999 Pai-Hsiang Hsiao.
24. * All rights reserved.
25. *
26. * Redistribution and use in source and binary forms, with or without
27. * modification, are permitted provided that the following conditions
28. * are met:
29. *
30. * . Redistributions of source code must retain the above copyright
31. * notice, this list of conditions and the following disclaimer.
32. * . Redistributions in binary form must reproduce the above copyright
33. * notice, this list of conditions and the following disclaimer in
34. * the documentation and/or other materials provided with the
35. * distribution.
36. * . Neither the name of the TaBE Project nor the names of its
37. * contributors may be used to endorse or promote products derived
38. * from this software without specific prior written permission.
39. *
40. * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
41. * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
42. * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
43. * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
44. * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
45. * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
46. * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
47. * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
48. * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
49. * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
50. * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
51. * OF THE POSSIBILITY OF SUCH DAMAGE.
52. */
53. /*
54. * Copyright (c) 1999 Computer Systems and Communication Lab,
55. * Institute of Information Science, Academia Sinica.
56. * All rights reserved.
57. *
58. * Redistribution and use in source and binary forms, with or without
59. * modification, are permitted provided that the following conditions
60. * are met:
61. *
62. * . Redistributions of source code must retain the above copyright
63. * notice, this list of conditions and the following disclaimer.
64. * . Redistributions in binary form must reproduce the above copyright
65. * notice, this list of conditions and the following disclaimer in
66. * the documentation and/or other materials provided with the
67. * distribution.

```
* . Neither the name of the Computer Systems and Communication Lab
* nor the names of its contributors may be used to endorse or
* promote products derived from this software without specific
* prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*/
Copyright 1996 Chih-Hao Tsai @ Beckman Institute, University of Illinois
c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
----------COPYING.libtabe---END----------------------------------
----------COPYING.ipadic---BEGIN---------------------------------
Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
and Technology. All Rights Reserved.
Use, reproduction, and distribution of this software is permitted.
Any copy of this software, whether in its original form or modified,
must include both the above copyright notice and the following
paragraphs.
Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.
A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.
Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.
NO WARRANTY
The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.
Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.
Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the
production of secondary products from the program.
In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
```

147. grants independently of ICOT any specific warranty to the user in
148. writing, such person, organization or entity, will also be exempted
149. from and not be held liable to the user for any such damages as noted
150. above as far as the program is concerned.
151. ----------COPYING.ipadic---END----------

3. Lao Word Break Dictionary Data (laodict.txt)

1. Copyright (c) 2013 International Business Machines Corporation
2. and others. All Rights Reserved.
   #
3. Project: http://code.google.com/p/lao-dictionary/
4. Dictionary: http://lao-dictionary.googlecode.com/git/Lao-Dictionary.txt
5. License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
6. (copied below)
   #
7. This file is derived from the above dictionary, with slight modifications.
8. ----------
9. Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
10. All rights reserved.
    #
11. Redistribution and use in source and binary forms, with or without modification,
12. are permitted provided that the following conditions are met:
    #
13. Redistributions of source code must retain the above copyright notice, this
14. list of conditions and the following disclaimer. Redistributions in binary
15. form must reproduce the above copyright notice, this list of conditions and
16. the following disclaimer in the documentation and/or other materials
17. provided with the distribution.
    #
18. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
19. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
20. WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
21. DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
22. ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
23. (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
24. LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
25. ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
26. (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
27. SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
28. ----------

4. Burmese Word Break Dictionary Data (burmesedict.txt)

1. Copyright (c) 2014 International Business Machines Corporation
2. and others. All Rights Reserved.
   #
3. This list is part of a project hosted at:
4. github.com/kanyawtech/myanmar-karen-word-lists
   #
5. ----------
6. Copyright (c) 2013, LeRoy Benjamin Sharon
7. All rights reserved.
   #
8. Redistribution and use in source and binary forms, with or without modification,
9. are permitted provided that the following conditions are met:
   #
10. Redistributions of source code must retain the above copyright notice, this
11. list of conditions and the following disclaimer.
    #
12. Redistributions in binary form must reproduce the above copyright notice, this
13. list of conditions and the following disclaimer in the documentation and/or
14. other materials provided with the distribution.
    #
15. Neither the name Myanmar Karen Word Lists, nor the names of its
16. contributors may be used to endorse or promote products derived from
17. this software without specific prior written permission.
    #
18. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
19. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
20. WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
21. DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
22. ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
23. (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
24. LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
25. ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
26. (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
27. SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
28. ----------

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

7. Database Ownership

The TZ database itself is not an IETF Contribution or an IETF
document. Rather it is a pre-existing and regularly updated work
that is in the public domain, and is intended to remain in the public
domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do not apply
to the TZ Database or contributions that individuals make to it.
Should any claims be made and substantiated against the TZ Database,
the organization that is providing the IANA Considerations defined in
this RFC, under the memorandum of understanding with the IETF,
currently ICANN, may act in accordance with all competent court
orders. No ownership claims will be made by ICANN or the IETF Trust
on the database or the code. Any person making a contribution to the
database or code waives all rights to future claims in that
contribution or in the TZ Database.

Boost Software License

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

libcurl License

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2017, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

libssh2 License

/* Copyright (c) 2004-2007 Sara Golemon <sarag@libssh2.org>
* Copyright (c) 2005,2006 Mikhail Gusarov <dottedmag@dottedmag.net>
* Copyright (c) 2006-2007 The Written Word, Inc.
* Copyright (c) 2007 Eli Fant <elifantu@mail.ru>
* Copyright (c) 2009-2014 Daniel Stenberg
* Copyright (C) 2008, 2009 Simon Josefsson
* All rights reserved.
*
* Redistribution and use in source and binary forms,
* with or without modification, are permitted provided
* that the following conditions are met:
*
* Redistributions of source code must retain the above
* copyright notice, this list of conditions and the
* following disclaimer.
*
* Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following
* disclaimer in the documentation and/or other materials
* provided with the distribution.
*
* Neither the name of the copyright holder nor the names
* of any other contributors may be used to endorse or
* promote products derived from this software without
* specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
* CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
* USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
* OF SUCH DAMAGE.
*/

Standard Function Library (SFL) License

This product uses parts of the iMatix SFL, Copyright © 1991-2000 iMatix Corporation <http://www.imatix.com>.

Python Software Foundation License Version 2

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
---

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby
grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce,
analyze, test, perform and/or display publicly, prepare derivative works,
distribute, and otherwise use Python alone or in any derivative version,
provided, however, that PSF's License Agreement and PSF's notice of copyright,
i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019 Python Software Foundation;
All Rights Reserved" are retained in Python alone or in any derivative version
prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on
or incorporates Python or any part thereof, and wants to make
the derivative work available to others as provided herein, then
Licensee hereby agrees to include in any such work a brief summary of
the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS"
basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any
relationship of agency, partnership, or joint venture between PSF and
Licensee. This License Agreement does not grant permission to use PSF
trademarks or trade name in a trademark sense to endorse or promote
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

Universal Enterprise Controller

xerces License

/*
* Copyright 1999-2004 The Apache Software Foundation.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Universal Enterprise Controller Client Applications

prefuse License

Copyright (c) 2004-2007 Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Universal Agent for SOA

This section contains license information for the following Universal Agent for SOA components:

- Axis2
- Tomcat
- DB Bery
- log4j

Each component's notice file contains information about external components used by a particular component.

All of these components are governed under the same Apache 2.0 license, below.

Apache 2.0 License

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright yyyy name of copyright owner

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE TOMCAT SUBCOMPONENTS:

Apache Tomcat includes a number of subcomponents with separate copyright notices and license terms. Your use of these subcomponents is subject to the terms and conditions of the following licenses.

For the Eclipse JDT Core Batch Compiler (ecj-x.x.x.jar) component:

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

For the Windows Installer component:

- All NSIS source code, plug-ins, documentation, examples, header files and graphics, with the exception of the compression modules and where otherwise noted, are licensed under the zlib/libpng license.
- The zlib compression module for NSIS is licensed under the zlib/libpng license.
- The bzip2 compression module for NSIS is licensed under the bzip2 license.
- The lzma compression module for NSIS is licensed under the Common Public License version 1.0.

zlib/libpng license

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

bzip2 license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@acm.org
Common Public License version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Special exception for LZMA compression module

Igor Pavlov and Amir Szekely, the authors of the LZMA compression module for NSIS, expressly permit you to statically or dynamically link your code (or bind by name) to the files from the LZMA compression module for NSIS without subjecting your linked code to the terms of the Common Public license version 1.0. Any modifications or additions to files from the LZMA compression module for NSIS, however, are subject to the terms of the Common Public License version 1.0.

For the following XML Schemas for Java EE Deployment Descriptors:

- javaee_5.xsd
- javaee_web_services_1_2.xsd
- javaee_web_services_client_1_2.xsd
- javaee_6.xsd
- javaee_web_services_1_3.xsd
- javaee_web_services_client_1_3.xsd
- jsp_2_2.xsd
- web-app_3_0.xsd
- web-common_3_0.xsd
- web-fragment_3_0.xsd
- javaee_7.xsd
- javaee_web_services_1_4.xsd
- javaee_web_services_client_1_4.xsd
- jsp_2_3.xsd
- web-app_3_1.xsd
- web-common_3_1.xsd
- web-fragment_3_1.xsd
- javaee_8.xsd
- web-app_4_0.xsd
- web-common_4_0.xsd
- web-fragment_4_0.xsd

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.
You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item,. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Apache Axis2 Notice

```
========================================================================
== NOTICE file corresponding to the section 4(d) of ==
== the Apache License, Version 2.0, ==
== in this case for the Apache Axis2 distribution. ==
========================================================================
```

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
Portions Copyright 2006 International Business Machines Corp.
Portions Copyright 2005-2007 WSO2, Inc.

This product also includes schemas and specification developed by:

- the W3C consortium (http://www.w3c.org)

This product also includes WS-* schemas developed by International Business Machines Corporation, Microsoft Corporation, BEA Systems, TIBCO Software, SAP AG, Sonic Software, and VeriSign

This product also includes a WSDL developed by salesforce.com

- Copyright 1999-2006 salesforce.com, inc.

Portions of the included xmlbeans library were originally based on the following:

- software copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

Please read the different LICENSE files present in the lib directory of this distribution.

Apache Tomcat Notice

Apache Tomcat
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code derived from netty-native developed by the Netty project (http://netty.io, https://github.com/netty/netty-tcnative/) and from finagle-native developed at Twitter (https://github.com/twitter/finagle).

The Windows Installer is built with the Nullsoft Scriptable Install System (NSIS), which is open source software. The original software and related information is available at http://nsis.sourceforge.net.

Java compilation software for JSP pages is provided by the Eclipse JDT Core Batch Compiler component, which is open source software. The original software and related information is available at http://www.eclipse.org/jdt/core/.

For portions of the Tomcat JNI OpenSSL API and the OpenSSL JSSE integration The org.apache.tomcat.jni and the org.apache.tomcat.net.openssl packages are derivative work originating from the Netty project and the finagle-native project developed at Twitter

- Copyright 2014 The Netty Project
- Copyright 2014 Twitter

The original XML Schemas for Java EE Deployment Descriptors:

- javaee_5.xsd
- javaee_web_services_1_2.xsd
- javaee_web_services_client_1_2.xsd
- javaee_6.xsd
- javaee_web_services_1_3.xsd
- javaee_web_services_client_1_3.xsd
- jsp_2_2.xsd
- web-app_3_0.xsd
- web-common_3_0.xsd
- web-fragment_3_0.xsd
- javaee_7.xsd
- javaee_web_services_1_4.xsd
- javaee_web_services_client_1_4.xsd
- jsp_2_3.xsd
- web-app_3_1.xsd
- web-common_3_1.xsd
- web-fragment_3_1.xsd

may be obtained from: http://www.oracle.com/webfolder/technetwork/jsc/xml/ns/javaee/index.html

Apache Derby Notice

```
=========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Derby distribution. ==
=========================================================================

Apache Derby
Copyright 2004-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Portions of Derby were originally developed by
International Business Machines Corporation and are
licensed to the Apache Software Foundation under the
"Software Grant and Corporate Contribution License Agreement",
informally known as the "Derby CLA".
The following copyright notice(s) were affixed to portions of the code
with which this file is now or was at one time distributed
and are placed here unaltered.

(C) Copyright 1997,2004 International Business Machines Corporation. All rights reserved.

(C) Copyright IBM Corp. 2003.

The portion of the functionTests under 'nist' was originally
developed by the National Institute of Standards and Technology (NIST),
an agency of the United States Department of Commerce, and adapted by
International Business Machines Corporation in accordance with the NIST
Software Acknowledgment and Redistribution document at http://www.itl.nist.gov/div897/ctg/sql_form.htm
```

Apache log4j Notice

```
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
```

Open Java Run Time Environment (Version 1.8)

OPENJDK ASSEMBLY EXCEPTION

```
The OpenJDK source code made available by Oracle America, Inc. (Oracle) at
openjdk.java.net ("OpenJDK Code") is distributed under the terms of the GNU
General Public License <http://www.gnu.org/copyleft/gpl.html> version 2
only ("GPL2"), with the following clarification and special exception.

Linking this OpenJDK Code statically or dynamically with other code
is making a combined work based on this library. Thus, the terms
and conditions of GPL2 cover the whole combination.
```

As a special exception, Oracle gives you permission to link this
OpenJDK Code with certain code licensed by Oracle as indicated at
http://openjdk.java.net/legal/exception-modules-2007-05-08.html
("Designated Exception Modules") to produce an executable,
regardless of the license terms of the Designated Exception Modules,
and to copy and distribute the resulting executable under GPL2,
provided that the Designated Exception Modules continue to be
governed by the licenses under which they were offered by Oracle.

As such, it allows licensees and sublicensees of Oracle's GPL2 OpenJDK Code
to build an executable that includes those portions of necessary code that
Oracle could not provide under GPL2 (or that Oracle has provided under GPL2
with the Classpath exception). If you modify or add to the OpenJDK code,
that new GPL2 code may still be combined with Designated Exception Modules
if the new code is made subject to this exception by its copyright holder.

The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license
document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share
and change it. By contrast, the GNU General Public License is intended to
guarantee your freedom to share and change free software--to make sure the
software is free for all its users. This General Public License applies to
most of the Free Software Foundation's software and to any other program whose
authors commit to using it. (Some other Free Software Foundation software is
covered by the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our
General Public Licenses are designed to make sure that you have the freedom to
distribute copies of free software (and charge for this service if you wish),
that you receive source code or can get it if you want it, that you can change
the software or use pieces of it in new free programs; and that you know you
can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny
you these rights or to ask you to surrender the rights. These restrictions
translate to certain responsibilities for you if you distribute copies of the
software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for
a fee, you must give the recipients all the rights that you have. You must
make sure that they, too, receive or can get the source code. And you must
show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2)
offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain that
everyone understands that there is no warranty for this free software. If the
software is modified by someone else and passed on, we want its recipients to
know that what they have is not the original, so that any problems introduced
by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We
wish to avoid the danger that redistributors of a free program will
individually obtain patent licenses, in effect making the program proprietary.
To prevent this, we have made it clear that any patent must be licensed for
everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification
follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are subject to the following clarification and special exception to the GPL, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

ASM Bytecode Manipulation Framework v5.0.3

%% This notice is provided with respect to ASM Bytecode Manipulation Framework v5.0.3, which may be included with JRE 8, and JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 2000-2011 France Télécom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

BSDiff v4.3

%% This notice is provided with respect to BSDiff v4.3, which may be
included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright 2003-2005 Colin Percival
All rights reserved

Redistribution and use in source and binary forms, with or without
modification, are permitted providing that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

CodeViewer 1.0

%% This notice is provided with respect to CodeViewer 1.0, which may be
included with JDK 8.

— begin of LICENSE —

Copyright 1999 by CoolServlets.com.

Any errors or suggested improvements to this class can be reported as
instructed on CoolServlets.com. We hope you enjoy this program... your
comments will encourage further development! This software is distributed
under the terms of the BSD License. Redistribution and use in source and
binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither name of CoolServlets.com nor the names of its contributors may be
used to endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COOLSERVLETS.COM AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

— end of LICENSE —

Cryptix AES 3.2.0

%% This notice is provided with respect to Cryptix AES 3.2.0, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Cryptix General License

Copyright (c) 1995-2005 The Cryptix Foundation Limited.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

CUP Parser Generator for Java 0.10k

%% This notice is provided with respect to CUP Parser Generator for Java 0.10k, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

— end of LICENSE —

DejaVu fonts v2.34

%% This notice is provided with respect to DejaVu fonts v2.34, which may be included with JRE 8, and JDK 8, and OpenJDK 8.

— begin of LICENSE —

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain. Glyphs imported from Arev fonts are (c) Tavmjong Bah (see below)

Bitstream Vera Fonts Copyright

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free . fr.

TeX Gyre DJV Math

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain.

Math extensions done by B. Jackowski, P. Strzelczyk and P. Pianowski (on behalf of TeX users groups) are in public domain.

Letters imported from Euler Fraktur from AMSfonts are (c) American Mathematical Society (see below).
Bitstream Vera Fonts Copyright
Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.
Except as contained in this notice, the names of GNOME, the GNOME Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the GNOME Foundation or Bitstream Inc., respectively.
For further information, contact: fonts at gnome dot org.

AMSFonts (v. 2.2) copyright

The PostScript Type 1 implementation of the AMSFonts produced by and previously distributed by Blue Sky Research and Y&Y, Inc. are now freely available for general use. This has been accomplished through the cooperation
of a consortium of scientific publishers with Blue Sky Research and Y&Y. Members of this consortium include:

Elsevier Science IBM Corporation Society for Industrial and Applied Mathematics (SIAM) Springer-Verlag American Mathematical Society (AMS)

In order to assure the authenticity of these fonts, copyright will be held by the American Mathematical Society. This is not meant to restrict in any way the legitimate use of the fonts, such as (but not limited to) electronic distribution of documents containing these fonts, inclusion of these fonts into other public domain or commercial font collections or computer applications, use of the outline data to create derivative fonts and/or faces, etc. However, the AMS does require that the AMS copyright notice be removed from any derivative versions of the fonts which have been altered in any way. In addition, to ensure the fidelity of TeX documents using Computer Modern fonts, Professor Donald Knuth, creator of the Computer Modern faces, has requested that any alterations which yield different font metrics be given a different name.

— end of LICENSE —

Document Object Model (DOM) Level 2 & 3

%% This notice is provided with respect to Document Object Model (DOM) Level 2 & 3, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

W3C SOFTWARE NOTICE AND LICENSE

http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1.The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

2.Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.

3.Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS,COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

— end of LICENSE —

Dynalink v0.5

%% This notice is provided with respect to Dynalink v0.5, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 2009-2013, Attila Szegedi

All rights reserved.Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Attila Szegedi nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THEPOSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

Elliptic Curve Cryptography

%% This notice is provided with respect to Elliptic Curve Cryptography, which may be included with JRE 8, JDK 8, and OpenJDK 8.

You are receiving a copy of the Elliptic Curve Cryptography library in source form with the JDK 8 and OpenJDK 8 source distributions, and as object code in the JRE 8 & JDK 8 runtimes.

In the case of the JRE & JDK runtimes, the terms of the Oracle license do NOT apply to the Elliptic Curve Cryptography library; it is licensed under the following license, separately from Oracle's JDK & JRE. If you do not wish to install the Elliptic Curve Cryptography library, you may delete the Elliptic Curve Cryptography library:

- On Solaris and Linux systems: delete $(JAVA_HOME)/lib/libsunec.so
- On Windows systems: delete $(JAVA_HOME)\bin\sunec.dll
- On Mac systems, delete:
  for JRE: /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/lib/libsunec.dylib
  for JDK: $(JAVA_HOME)/jre/lib/libsunec.dylib

Written Offer for ECC Source Code
For third party technology that you receive from Oracle in binary form which is licensed under an open source license that gives you the right to receive the source code for that binary, you can obtain a copy of the applicable source code from this page:
http://hg.openjdk.java.net/jdk8u/jdk8u/jdk/file/tip/src/share/native/sun/security/ec/impl

If the source code for the technology was not provided to you with the binary, you can also receive a copy of the source code on physical media by submitting a written request to:

Oracle America, Inc.
Attn: Associate General Counsel,
Development and Engineering Legal
500 Oracle Parkway, 10th Floor
Redwood Shores, CA 94065

Or, you may send an email to Oracle using the form at:
http://www.oracle.com/goto/opensourcecode/request

Your request should include:

- The name of the component or binary file(s) for which you are requesting the source code
- The name and version number of the Oracle product containing the binary
- The date you received the Oracle product
- Your name
- Your company name (if applicable)
- Your return mailing address and email and
- A telephone number in the event we need to reach you.

We may charge you a fee to cover the cost of physical media and processing. Your request must be sent (i) within three (3) years of the date you received the Oracle product that included the component or binary file(s) that are the subject of your request, or (ii) in the case of code licensed under the GPL v3, for as long as Oracle offers spare parts or customer support for that product model.

— begin of LICENSE —

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages – typically libraries – of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

— end of LICENSE —

ECMAScript Language Specification ECMA-262 Edition 5.1

%% This notice is provided with respect to ECMAScript Language
Specification ECMA-262 Edition 5.1, which may be included with
JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright notice
Copyright © 2011 Ecma International
Ecma International
Rue du Rhone 114
CH-1204 Geneva
Tel: +41 22 849 6000
Fax: +41 22 849 6001
Web: http://www.ecma-international.org

This document and possible translations of it may be copied and furnished to
others, and derivative works that comment on or otherwise explain it or assist
in its implementation may be prepared, copied, published, and distributed, in
whole or in part, without restriction of any kind, provided that the above
copyright notice and this section are included on all such copies and derivative
works. However, this document itself may not be modified in any way, including
by removing the copyright notice or references to Ecma International, except as
needed for the purpose of developing any document or deliverable produced by
Ecma International (in which case the rules applied to copyrights must be
followed) or as required to translate it into languages other than English. The
limited permissions granted above are perpetual and will not be revoked by Ecma
International or its successors or assigns. This document and the information
contained herein is provided on an "AS IS" basis and ECMA INTERNATIONAL
DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP
RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR
PURPOSE." Software License

All Software contained in this document ("Software)" is protected by copyright
and is being made available under the "BSD License", included below. This
Software may be subject to third party rights (rights from parties other than
Ecma International), including patent rights, and no licenses under such third
party rights are granted under this license even if the third party concerned is
a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS
AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*. Redistribution and use in source and
binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

3. Neither the name of the authors nor Ecma International may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
— end of LICENSE —

freebXML Registry 3.0 & 3.1

%% This notice is provided with respect to freebXML Registry 3.0 & 3.1, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

freebxml: Copyright (c) 2001 freebxml.org. All rights reserved.

The names "The freebXML Registry Project" and "freebxml Software Foundation" must not be used to endorse or promote products derived from this software or be used in a product name without prior written permission. For written permission, please contact ebxmlrr-team@lists.sourceforge.net.

This software consists of voluntary contributions made by many individuals on behalf of the the freebxml Software Foundation. For more information on the freebxml Software Foundation, please see <http://www.freebxml.org/>.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

The freebxml License, Version 1.1 5
Copyright (c) 2001 freebxml.org. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowlegement:
"This product includes software developed by freebxml.org (http://www.freebxml.org/)."
Alternately, this acknowlegement may appear in the software itself, if and wherever such third-party acknowlegements normally appear.

4. The names "The freebXML Registry Project", "freebxml Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact ebxmlrr-team@lists.sourceforge.net.

5. Products derived from this software may not be called "freebxml", "freebXML Registry" nor may freebxml" appear in their names without prior written permission of the freebxml Group.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE freebxml SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

IAIK PKCS#11 Wrapper

%% This notice is provided with respect to IAIK PKCS#11 Wrapper,
which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

IAIK PKCS#11 Wrapper License

Copyright (c) 2002 Graz University of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must
include the following acknowledgment:

"This product includes software developed by IAIK of Graz University of
Technology."

Alternately, this acknowledgment may appear in the software itself, if and
wherever such third-party acknowledgments normally appear.

4. The names "Graz University of Technology" and "IAIK of Graz University of
Technology" must not be used to endorse or promote products derived from this
software without prior written permission.

5. Products derived from this software may not be called "IAIK PKCS Wrapper",
nor may "IAIK" appear in their name, without prior written permission of
Graz University of Technology.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

ICU4C 4.0.1 and ICU4J 4.4

%% This notice is provided with respect to ICU4C 4.0.1 and ICU4J 4.4, which
may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 1995-2010 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, and/or sell copies of the
Software, and to permit persons to whom the Software is furnished to do so,
provided that the above copyright notice(s) and this permission notice appear
in all copies of the Software and that both the above copyright notice(s) and
this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.
All trademarks and registered trademarks mentioned herein are the property of their respective owners.

— end of LICENSE —

IJG JPEG 6b

%% This notice is provided with respect to IJG JPEG 6b, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:
(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

— end of LICENSE —

Jing 20030619

%% This notice is provided with respect to Jing 20030619, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 2001-2003 Thai Open Source Software Center Ltd All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Thai Open Source Software Center Ltd nor
the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

— end of LICENSE —

Joni v1.1.9

%% This notice is provided with respect to Joni v1.1.9, which may be
included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

— end of LICENSE —

JOpt-Simple v3.0

%% This notice is provided with respect to JOpt-Simple v3.0, which may be
included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 2004-2009 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

— end of LICENSE —

Kerberos functionality

%% This notice is provided with respect to Kerberos functionality, which
which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

(C) Copyright IBM Corp. 1999 All Rights Reserved.
Copyright 1997 The Open Group Research Institute. All rights reserved.

— end of LICENSE —

Kerberos functionality from FundsXpress, INC.

%% This notice is provided with respect to Kerberos functionality from
FundsXpress, INC., which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (C) 1998 by the FundsXpress, INC.

All rights reserved.

Export of this software from the United States of America may require
a specific license from the United States Government. It is the
responsibility of any person or organization contemplating export to
obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and
distribute this software and its documentation for any purpose and
without fee is hereby granted, provided that the above copyright
notice appear in all copies and that both that copyright notice and
this permission notice appear in supporting documentation, and that
the name of FundsXpress. not be used in advertising or publicity pertaining
to distribution of the software without specific, written prior
permission. FundsXpress makes no representations about the suitability of
this software for any purpose. It is provided "as is" without express
or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

— end of LICENSE —

Kronos OpenGL headers

%% This notice is provided with respect to Kronos OpenGL headers, which may be
included with JDK 8 and OpenJDK 8 source distributions.

— begin of LICENSE —

Copyright (c) 2007 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and/or associated documentation files (the "Materials"), to
deal in the Materials without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Materials, and to permit persons to whom the Materials are
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE
MATERIALS.

— end of LICENSE —

Portions Copyright Eastman Kodak Company

%% Portions Copyright Eastman Kodak Company 1991-2003

%% This notice is provided with respect to libpng 1.6.35, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.

libpng versions 1.0.7, July 1, 2000 through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu
Mandar Sahastrabuddhe
Google Inc.
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

END OF COPYRIGHT NOTICE, DISCLAIMER, and LICENSE.

TRADEMARK:

The name "libpng" has not been registered by the Copyright owner as a trademark in any jurisdiction. However, because libpng has been distributed and maintained world-wide, continually since 1995, the Copyright owner claims "common-law trademark protection" in any jurisdiction where common-law trademark is recognized.

OSI CERTIFICATION:

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative. OSI has not addressed the additional disclaimers inserted at version 1.0.7.

EXPORT CONTROL:

The Copyright owner believes that the Export Control Classification Number (ECCN) for libpng is EAR99, which means not subject to export controls or International Traffic in Arms Regulations (ITAR) because it is open source, publicly available software, that does not contain any encryption software. See the EAR, paragraphs 734.3(b)(3) and 734.7(b).

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
July 15, 2018

— end of LICENSE —

GIFLIB 5.1.1 & libungif 4.1.3

%% This notice is provided with respect to GIFLIB 5.1.1 & libungif 4.1.3, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

— end of LICENSE —

Little CMS 2.9

%% This notice is provided with respect to Little CMS 2.9, which may be
included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Little CMS
Copyright (c) 1998-2011 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

— end of LICENSE —

Mesa 3D Graphics Library v4.1

%% Lucida is a registered trademark or trademark of Bigelow & Holmes in the
U.S. and other countries.

%% This notice is provided with respect to Mesa 3D Graphics Library v4.1,
which may be included with JRE 8, JDK 8, and OpenJDK 8 source distributions.

— begin of LICENSE —

Mesa 3-D graphics library
Version: 4.1

Copyright (C) 1999-2002 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

— end of LICENSE —

Mozilla Network Security Services (NSS)

%% This notice is provided with respect to Mozilla Network Security Services (NSS), which is supplied with the JDK test suite in the OpenJDK source code repository. It is licensed under Mozilla Public License (MPL), version 2.0.

The NSS libraries are supplied in executable form, built from unmodified NSS source code labeled with the "NSS_3_16_RTM" HG tag.

The NSS source code is available in the OpenJDK source code repository at: jdk/test/sun/security/pkcs11/nss/src

The NSS libraries are available in the OpenJDK source code repository at: jdk/test/sun/security/pkcs11/nss/lib

— begin of LICENSE —

Mozilla Public License Version 2.0
==========

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

****

- *
- 6. Disclaimer of Warranty *
- ---------- *
- *
- Covered Software is provided under this License on an "as is" *
- basis, without warranty of any kind, either expressed, implied, or *
- statutory, including, without limitation, warranties that the *
- Covered Software is free of defects, merchantable, fit for a *
- particular purpose or non-infringing. The entire risk as to the *
- quality and performance of the Covered Software is with You. *
- Should any Covered Software prove defective in any respect, You *
- (not any Contributor) assume the cost of any necessary servicing, *
- repair, or correction. This disclaimer of warranty constitutes an *
- essential part of this License. No use of any Covered Software is *
- authorized under this License except under this disclaimer. *
- *
  ****

****

- *
- 7. Limitation of Liability *
- ---------- *
- *

- Under no circumstances and under no legal theory, whether tort *
- (including negligence), contract, or otherwise, shall any *
- Contributor, or anyone who distributes Covered Software as *
- permitted above, be liable to You for any direct, indirect, *
- special, incidental, or consequential damages of any character *
- including, without limitation, damages for lost profits, loss of *
- goodwill, work stoppage, computer failure or malfunction, or any *
- and all other commercial damages or losses, even if such party *
- shall have been informed of the possibility of such damages. This *
- limitation of liability shall not apply to liability for death or *
- personal injury resulting from such party's negligence to the *
- extent applicable law prohibits such limitation. Some *
- jurisdictions do not allow the exclusion or limitation of *
- incidental or consequential damages, so this exclusion and *
- limitation may not apply to You. *
- *
  ****

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular
file, then You may include the notice in a location (such as a LICENSE
file in a relevant directory) where a recipient would be likely to look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as
defined by the Mozilla Public License, v. 2.0.

— end of LICENSE —

PC/SC Lite for Suse Linux v.1.1.1

%% This notice is provided with respect to PC/SC Lite for Suse Linux v.1.1.1,
which may be included with JRE 8, JDK 8, and OpenJDK 8 on Linux and Solaris.

— begin of LICENSE —

Copyright (c) 1999-2004 David Corcoran <corcoran@linuxnet.com>
Copyright (c) 1999-2004 Ludovic Rousseau <ludovic.rousseau (at) free.fr>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
must display the following acknowledgement:
This product includes software developed by:
David Corcoran <corcoran@linuxnet.com>
http://www.linuxnet.com (MUSCLE)
4. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

Changes to this license can be made only by the copyright author with
explicit written consent.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

PorterStemmer v4

%% This notice is provided with respect to PorterStemmer v4, which may be
included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

See: http://tartarus.org/~martin/PorterStemmer

The software is completely free for any purpose, unless notes at the head of
the program text indicates otherwise (which is rare). In any case, the notes
about licensing are never more restrictive than the BSD License.

In every case where the software is not written by me (Martin Porter), this
licensing arrangement has been endorsed by the contributor, and it is
therefore unnecessary to ask the contributor again to confirm it.

I have not asked any contributors (or their employers, if they have them) for
proofs that they have the right to distribute their software in this way.

— end of LICENSE —

Relax NG Object/Parser v.20050510

%% This notice is provided with respect to Relax NG Object/Parser v.20050510, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) Kohsuke Kawaguchi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

— end of LICENSE —

RelaxNGCC v1.12

%% This notice is provided with respect to RelaxNGCC v1.12, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 2000-2003 Daisuke Okajima and Kohsuke Kawaguchi.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (http://relaxngcc.sf.net/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names of the copyright holders must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact the copyright holders.

5. Products derived from this software may not be called "RELAXNGCC", nor may "RELAXNGCC" appear in their name, without prior written permission of the copyright holders.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

Relax NG Datatype 1.0

%% This notice is provided with respect to Relax NG Datatype 1.0, which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (c) 2005, 2010 Thai Open Source Software Center Ltd
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the names of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— end of LICENSE —

SoftFloat version 2b

%% This notice is provided with respect to SoftFloat version 2b, which may be included with JRE 8, JDK 8, and OpenJDK 8 on Linux/ARM.

— begin of LICENSE —

Use of any of this software is governed by the terms of the license below:

SoftFloat was written by me, John R. Hauser. This work was made possible in part by the International Computer Science Institute, located at Suite 600, 1947 Center Street, Berkeley, California 94704. Funding was partially provided by the National Science Foundation under grant MIP-9311980. The original version of this code was written as part of a project to build a fixed-point vector processor in collaboration with the University of California at Berkeley, overseen by Profs. Nelson Morgan and John Wawrzynek.

THIS SOFTWARE IS DISTRIBUTED AS IS, FOR FREE. Although reasonable effort has been made to avoid it, THIS SOFTWARE MAY CONTAIN FAULTS THAT WILL AT TIMES RESULT IN INCORRECT BEHAVIOR. USE OF THIS SOFTWARE IS RESTRICTED TO PERSONS AND ORGANIZATIONS WHO CAN AND WILL TAKE FULL RESPONSIBILITY FOR ALL LOSSES, COSTS, OR OTHER PROBLEMS THEY INCUR DUE TO THE SOFTWARE, AND WHO FURTHERMORE EFFECTIVELY INDEMNIFY JOHN HAUSER AND THE INTERNATIONAL COMPUTER SCIENCE INSTITUTE (possibly via similar legal warning) AGAINST ALL LOSSES, COSTS, OR OTHER PROBLEMS INCURRED BY THEIR CUSTOMERS AND CLIENTS DUE TO THE SOFTWARE.

Derivative works are acceptable, even for commercial purposes, provided that the minimal documentation requirements stated in the source code are satisfied.

— end of LICENSE —

Sparkle 1.5

%% This notice is provided with respect to Sparkle 1.5,
which may be included with JRE 8 on Mac OS X.

— begin of LICENSE —

Copyright (c) 2012 Sparkle.org and Andy Matuschak

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

— end of LICENSE —

Thai Dictionary

%% Portions licensed from Taligent, Inc.

%% This notice is provided with respect to Thai Dictionary, which may be
included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Copyright (C) 1982 The Royal Institute, Thai Royal Government.

Copyright (C) 1998 National Electronics and Computer Technology Center,
National Science and Technology Development Agency,
Ministry of Science Technology and Environment,
Thai Royal Government.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

— end of LICENSE —

Unicode 6.2.0 & CLDR 21.0.1

%% This notice is provided with respect to Unicode 6.2.0 & CLDR 21.0.1,
which may be included with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

Unicode Terms of Use

For the general privacy policy governing access to this site, see the Unicode
Privacy Policy. For trademark usage, see the Unicode® Consortium Name and
Trademark Usage Policy.

A. Unicode Copyright.
1. Copyright © 1991-2013 Unicode, Inc. All rights reserved.

2. Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode® Standard, subject to Terms and Conditions herein.

3. Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files solely for informational purposes in the creation of products supporting the Unicode Standard, subject to the Terms and Conditions herein.

4. Further specifications of rights and restrictions pertaining to the use of the particular set of data files known as the "Unicode Character Database" can be found in Exhibit 1.

5. Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions (Unicode 5.0 and earlier), these are found on the back of the title page. The online code charts carry specific restrictions. All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

6. No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

7. Modification is not permitted with respect to this document. All copies of this document must be verbatim.

B. Restricted Rights Legend. Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

C. Warranties and Disclaimers.
1. This publication and/or website may include technical or typographical errors or other inaccuracies . Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

2. If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

3. EXCEPT AS PROVIDED IN SECTION C.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

D. Waiver of Damages. In no event shall Unicode or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

E.Trademarks & Logos.
1. The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. “The Unicode Consortium” and “Unicode, Inc.” are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.’s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

2. The Unicode Consortium Name and Trademark Usage Policy (“Trademark Policy”) are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

3. All third party trademarks referenced herein are the property of their respective owners.

Miscellaneous.
1. Jurisdiction and Venue. This server is operated from a location in the State of California, United States of America. Unicode makes no representation that the materials are appropriate for use in other locations. If you access this server from other locations, you are responsible for compliance with local laws. This Agreement, all use of this site and any claims and damages resulting from use of this site are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this site shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

2. Modification by Unicode. Unicode shall have the right to modify this Agreement at any time by posting it to this site. The user may not assign any part of this Agreement without Unicode’s prior written consent.

3. Taxes. The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode’s net income.

4. Severability. If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

5. Entire Agreement. This Agreement constitutes the entire agreement between the parties.

EXHIBIT 1
UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/. Unicode Data Files do not include PDF online code charts under the directory http://www.unicode.org/Public/.
Software includes any source code published in the Unicode Standard or under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2012 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc. in the United States and other countries. All third party trademarks referenced herein are the property of their respective owners.

— end of LICENSE —

UPX v3.01

%% This notice is provided with respect to UPX v3.01, which may be included with JRE 8 on Windows.

— begin of LICENSE —

Use of any of this software is governed by the terms of the license below:

```
ooooo ooo ooooooooo. oooooooo ooooo
`888' `8' `888 `Y88. `8888 d8'
888 8 888 .d88' Y888..8P
888 8 888ooo88P' `8888'
888 8 888 .8PY888.
`88. .8' 888 d8' `888b
`YbodP' o888o o888o o888888o
```

The Ultimate Packer for eXecutables
Copyright (c) 1996-2000 Markus Oberhumer & Laszlo Molnar
http://wildsau.idv.uni-linz.ac.at/mfx/upx.html
http://www.nexus.hu/upx
http://upx.tsx.org

PLEASE CAREFULLY READ THIS LICENSE AGREEMENT, ESPECIALLY IF YOU PLAN TO MODIFY THE UPX SOURCE CODE OR USE A MODIFIED UPX VERSION.

ABSTRACT
========

UPX and UCL are copyrighted software distributed under the terms of the GNU General Public License (hereinafter the "GPL").

The stub which is imbedded in each UPX compressed program is part of UPX and UCL, and contains code that is under our copyright. The terms of the GNU General Public License still apply as compressing a program is a special form of linking with our stub.

As a special exception we grant the free usage of UPX for all executables, including commercial programs.
See below for details and restrictions.

COPYRIGHT
=========

UPX and UCL are copyrighted software. All rights remain with the authors.

UPX is Copyright (C) 1996-2000 Markus Franz Xaver Johannes Oberhumer
UPX is Copyright (C) 1996-2000 Laszlo Molnar

UCL is Copyright (C) 1996-2000 Markus Franz Xaver Johannes Oberhumer

GNU GENERAL PUBLIC LICENSE
==========================

UPX and the UCL library are free software; you can redistribute them
and/or modify them under the terms of the GNU General Public License as
published by the Free Software Foundation; either version 2 of
the License, or (at your option) any later version.

UPX and UCL are distributed in the hope that they will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; see the file COPYING.

SPECIAL EXCEPTION FOR COMPRESSED EXECUTABLES
==========

The stub which is imbedded in each UPX compressed program is part
of UPX and UCL, and contains code that is under our copyright. The
terms of the GNU General Public License still apply as compressing
a program is a special form of linking with our stub.

Hereby Markus F.X.J. Oberhumer and Laszlo Molnar grant you special
permission to freely use and distribute all UPX compressed programs
(including commercial ones), subject to the following restrictions:

1. You must compress your program with a completely unmodified UPX
version; either with our precompiled version, or (at your option)
with a self compiled version of the unmodified UPX sources as
distributed by us.
2. This also implies that the UPX stub must be completely unmodfied, i.e.
the stub imbedded in your compressed program must be byte-identical
to the stub that is produced by the official unmodified UPX version.
3. The decompressor and any other code from the stub must exclusively get
used by the unmodified UPX stub for decompressing your program at
program startup. No portion of the stub may get read, copied,
called or otherwise get used or accessed by your program.

ANNOTATIONS
==========

- You can use a modified UPX version or modified UPX stub only for programs that are compatible with the GNU General Public License.
- We grant you special permission to freely use and distribute all UPX compressed programs. But any modification of the UPX stub (such as, but not limited to, removing our copyright string or making your program non-decompressible) will immediately revoke your right to use and distribute a UPX compressed program.
- UPX is not a software protection tool; by requiring that you use the unmodified UPX version for your proprietary programs we make sure that any user can decompress your program. This protects both you and your users as nobody can hide malicious code - any program that cannot be decompressed is highly suspicious by definition.
- You can integrate all or part of UPX and UCL into projects that are compatible with the GNU GPL, but obviously you cannot grant any special exceptions beyond the GPL for our code in your project.
- We want to actively support manufacturers of virus scanners and similar security software. Please contact us if you would like to incorporate parts of UPX or UCL into such a product.

Markus F.X.J. Oberhumer Laszlo Molnar
markus.oberhumer@jk.uni-linz.ac.at ml1050@cdata.tvnet.hu

Linz, Austria, 25 Feb 2000

Additional License(s)

The UPX license file is at http://upx.sourceforge.net/upx-license.html.

— end of LICENSE —

Xfree86-VidMode Extension 1.0

%% This notice is provided with respect to Xfree86-VidMode Extension 1.0,
which may be included with JRE 8, JDK 8, and OpenJDK 8 on Linux and Solaris.

— begin of LICENSE —

Version 1.1 of XFree86 ProjectLicence.

Copyright (C) 1994-2004 The XFree86 Project, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicence, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished
to do so,subject to the following conditions:

1. Redistributions of source code must retain the above copyright
notice,this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution, and in the same place
and form as other copyright, license and disclaimer information.

3. The end-user documentation included with the redistribution, if any,must
include the following acknowledgment: "This product includes
software developed by The XFree86 Project, Inc (http://www.xfree86.org/) and
its contributors", in the same place and form as other third-party
acknowledgments. Alternately, this acknowledgment may appear in the software
itself, in the same form and location as other such third-party
acknowledgments.

4. Except as contained in this notice, the name of The XFree86 Project,Inc
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization from
The XFree86 Project, Inc.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES,INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE XFREE86 PROJECT, INC OR ITS CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL,SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO,PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

— end of LICENSE —

X Window System 6.8.2

%% This notice is provided with respect to X Window System 6.8.2, which may be
included with JRE 8, JDK 8, and OpenJDK 8 on Linux and Solaris.

— begin of LICENSE —

This is the copyright for the files in src/solaris/native/sun/awt: list.h,
multiVis.h, wsutils.h, list.c, multiVis.c
Copyright (c) 1994 Hewlett-Packard Co.
Copyright (c) 1996 X Consortium

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization
from the X Consortium.

The files in motif/lib/Xm/util included this copyright:mkdirhier.man,
xmkmf.man, chownxterm.c, makeg.man, mergelib.cpp, lndir.man, makestrs.man,
checktree.c, lndir.c, makestrs.c
Copyright (c) 1993, 1994 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not
be used in advertising or otherwise to promote the sale, use or other
dealing in this Software without prior written authorization from the
X Consortium.

Xmos_r.h:
/*
Copyright (c) 1996 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from the X Consortium.
*/

Copyright notice for HPkeysym.h:
/*

Copyright 1987, 1998 The Open Group

All Rights Reserved.

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization
from The Open Group.

Copyright 1987 by Digital Equipment Corporation, Maynard, Massachusetts,

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the names of Hewlett Packard
or Digital not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD
TO THIS SOFWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. Hewlett-Packard shall not be liable for errors
contained herein or direct, indirect, special, incidental or
consequential damages in connection with the furnishing,
performance, or use of this material.

*/

Copyright notice in keysym2ucs.h:

Copyright 1987, 1994, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation.

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization
from The Open Group.

Copyright 1987 by Digital Equipment Corporation, Maynard, Massachusetts

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of Digital not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

*/

— end of LICENSE —

zlib v1.2.11

%% This notice is provided with respect to zlib v1.2.11, which may be included
with JRE 8, JDK 8, and OpenJDK 8.

— begin of LICENSE —

version 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

— end of LICENSE —

Apache

%% This notice is provided with respect to the following which may be
included with JRE 8, JDK 8, and OpenJDK 8.

Apache Commons Math 3.2
Apache Derby 10.11.1.2
Apache Jakarta BCEL 5.1
Apache Jakarta Regexp 1.4
Apache Santuario XML Security for Java 1.5.4
Apache Xalan-Java 2.7.1
Apache Xerces Java 2.10.0
Apache XML Resolver 1.1

— begin of LICENSE —

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

— end of LICENSE —

Universal Agent 7.2.x Installation Requirements and Summary

- Universal Agent Components
- Platform Requirements
- Space Requirements
- Network Requirements
  - Ports Configuration
- Additional Requirements

Universal Agent Components

Universal Agent is comprised of a suite of components that are installed as a single installation package.

There is a separate package for each of the Universal Agent supported platforms. There also are additional, separate packages for Windows and UNIX.

The following table provides links to the installation requirements and summary (including any customization information) for the packages on each platform:

Platforms	Packages
z/OS	Universal Agent 7.2.x for z/OS
Windows	Universal Agent 7.2.x for Windows Universal Enterprise Controller 7.2.x Universal Enterprise Controller Client Applications 7.2.x Universal Agent for SOA 7.2.x
UNIX	Universal Agent 7.2.x for UNIX Universal Agent for SOA 7.2.x
IBM i	Workload Automation 5.1.1 for IBM i

For complete instructions on product installation and customization, see the Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Space Requirements

Space requirements for Universal Agent are driven largely by logging. A minimum 500MB of space is recommended. Each Agent contains a Universal Broker log and an Agent log.

The Universal Broker LOG_FILE_GENERATIONS configuration option lets you specify how many log files to save in the Universal Broker log directory. The default is 5.

The Agent log rolls over at midnight (12:00 a.m.) or whenever the Agent is restarted. The Log File Retention Period in Days Universal Controller system property lets you specify the number of days that an Agent log file (in addition to a Controller log file) is retained before it is purged. The default is 5 days.

Network Requirements

Universal Agent components run on z/OS, Windows, UNIX, and IBM i operating systems. These systems must be connected with a network that supports TCP/IP. For example, the Universal Command Manager running on a z/OS system must be able to establish a TCP/IP socket connection with the Universal Command Server running on a UNIX or Windows system.

TCP/IP provides a set of commands to verify network connectivity between systems. For example, the ping command can determine if an IP connection is possible between two systems. However, the ping command may not work in all network environments. In addition, the ping command may work, but a firewall may deny all other connection attempts between the two systems. Check with your local network administrator to determine the capabilities and topology of your network.

Universal Agent offers configuration parameters that can facilitate connections through network firewalls. Due to the large variety of firewall configurations, all possibilities cannot be discussed in this document. Check with your local network administrator to determine if a firewall is between the computer systems involved.

Ports Configuration

Universal Agent is composed of several components that utilize the TCP/IP network for component communications. TCP/IP routes must be available between the distributed components and firewall rules open, if necessary. The primary components using TCP/IP are Universal Broker, Universal Managers, and Universal Automation Center Agent (UAG).

- Universal Broker executes on all servers on which workload management services are required. The Universal Broker accepts TCP socket connections on port 7887 from remote clients such as Universal Command Manager and Universal Data Mover Manager. A server running Universal Agent will need to be able to accept TCP connections on port 7887 in order to provide such services.
- Universal Manager components include Universal Command Manager, Universal Data Mover Manager, and others. Manager components initiate TCP socket connections to remote Universal Brokers. If Universal Manager services are required, the Managers will need to be able to open TCP connections to remote Universal Brokers on port 7887.
- UAG initiates a TCP socket connection to the Universal Message Service (OMS) deployed on a remote, centrally located server. OMS accepts TCP connections on port 7878. UAG must be able to open a TCP connection to the remote OMS server on port 7878.

The port numbers above are default port numbers. They can be changed in the respective component configuration, if necessary.

Additional Requirements

In order to install any of these Universal Agent components, you must be able to write to the directory from which the installation is launched.

Installation Requirements and Summary - z/OS

- Objective
- System Requirements
- Platform Requirements
- Data Set Space Requirements
- Installation Summary
    - Customization

Objective

This page provides installation requirements and summary for Universal Agent 7.2.x on a z/OS operating system.

System Requirements

Universal Agent for z/OS requires the following software releases:

- z/OS 2.2, 2.3, or 2.4.
- SMP/E 3.5 or later.
- IBM Communication Server for z/OS 2.2, 2.3, or 2.4.
- IBM Language Environment (LE) for z/OS 2.2, 2.3, or 2.4.
- Windows workstation capable of establishing a TCP/IP network connection to the z/OS system.
- TSO user ID with an OMVS segment.
- About 1900 cylinders of DASD.
- Two available TCP/IP ports on z/OS.

All Universal Agent programs use z/OS UNIX System Services. As such, z/OS UNIX requires the user profile with which a program executes to have a properly defined OMVS segment. The OMVS segment should define a unique UID value. The HOME value must specify an existing home directory to which the user ID has read and write access.

Additionally, the group(s) that the user ID is associated with must have an OMVS segment that defines a unique GID value for the group. Refer to IBM's UNIX System Services Planning manual for additional details on defining z/OS UNIX users.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Data Set Space Requirements

As part of the Universal Agent for z/OS package installation, a number of SMP/E and non-SMP/E data sets are allocated and cataloged. The space requirements for these data sets are listed in z/OS Installation - Data Set Inventory.

Installation Summary

Step 1	Download one of the Universal Agent 7.2.x for z/OS product distribution files.
Step 2a	If one of the distribution files with the .tar.Z extension was downloaded, decompress it using any utility that can process files in Lempel-Ziv (that is, **compress**) format. Use **tar** to extract the files from the uncompressed file. To do both in a single statement, decompress the file to stdout and pipe the result to **tar** (for example, **zcat sb-v.r.m.l-zos.tar.Z | tar xvf -)**.
Step 2b	If one of the distribution files with the .zip extension was downloaded, decompress it using any utility that can process files in **ZIP** (or, PKZIP) format. The .zip file can also be extracted using Windows native file extraction tools.
Step 3	Modify the UNVFTP.TXT script provided by the distribution file. This script contains statements to upload the installation files and sample JCL, which can be used to process the files after upload. Two versions of these files are included in the distribution file. The files in the main directory are EBCDIC-encoded files. ASCII-encoded versions of these files are in the `./ascii` subdirectory. See z/OS Installation - Transferring Installation Files to z/OS for complete details.
Step 4	Modify the JCL uploaded from UNVRECV.JCL and run the job to receive the TSO XMIT file(s). If the distribution file that contains the DFDSS archive is used, UNVRECV.JCL also contains a step to restore the contents of the archive. See Receive the Installation Datasets for complete details.

Step 5	Edit the INSTALL(#SETUP) member to meet local requirements as described in the #SETUP comment prolog. #SETUP creates customized installation JCL members in the INSTALL library. Execute the installation jobs as required for your SMP/E environment. See z/OS Installation - SMPE Installation for complete details.
Step 6	Create started task user and group profiles for Universal Broker and Universal Enterprise Controller. See z/OS Configuration - Started Tasks for complete details.
Step 7	Define the SUNVLOAD library to APF, Program Control, and LNKLST. See z/OS Configuration - Load Library for complete details.
Step 8	Define SMF exits for Universal Command and Universal Automation Center Agent. See z/OS Configuration - SMF Exits for complete details.
Step 9	Configure the UNIX System Services file systems used by the Universal Broker and Universal Enterprise Controller. See Universal Agent Database Configuration for complete details.
Step 10	Perform individual component customization, including JCL procedure edits and adding component licenses. See z/OS Installation - Customization for complete details.
Step 11	Follow the optional steps described in z/OS USS Installation to complete the installation, if desired.

Customization

Customization consists of editing PDS members in the Universal configuration library, UNVCONF.

Installation Requirements and Summary - Windows

- Objective
- Universal Agent - System Installation
  - Windows Versions
  - Additional Requirements
  - Platform Requirements
  - Installation Account
  - Universal Broker Service
  - Spool Directory
- Universal Agent - User Mode Installation
  - Windows Versions
  - Additional Requirements
  - Platform Requirements
  - Installation Account
  - Universal Broker Service
  - Selecting a Universal Broker Service Account
  - Spool Directory
- Universal Enterprise Controller (UEC)
  - Windows Versions
  - Additional Requirements
- Universal Enterprise Controller Client Applications
  - Windows Versions
  - Additional Requirements
  - Java Runtime Environment
- Universal Agent for SOA
  - Additional Requirements
- Installation Summary
  - Customization

Objective

This page provides installation requirements and summary for Universal Agent 7.2.x on a Windows operating system.

Universal Agent for Windows is comprised of one required package - Universal Agent - and three optional packages: Universal Enterprise Controller, Universal Enterprise Controller Client Applications, and Universal Agent for SOA.

Universal Agent - System Installation

Windows Versions

To install Universal Agent for Windows, you must have one of the following versions of Windows:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

Additional Requirements

- An account with administrative privileges.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 325 megabytes of disk space.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Installation Account

The account used to install the package must have administrative privileges.

Universal Broker Service

The Universal Broker service runs either with an Administrative account or with the Local System account.

If you are using an Administrative account, the account must have the following privileges:

- Act as part of the operating system
- Adjust memory quotas for a process
- Bypass traverse checking
- Debug programs
- Deny log on as batch job
- Deny log on locally
- Deny log on through Terminal Services
- Log on as a service
- Impersonate a client after authentication
- Increase scheduling priority
- Replace a process level token
- Take ownership of files and other objects

UBrokerService: The Default Universal Broker Service Account

The Universal Agent for Windows installation will create an account with the privileges listed above, if the account does not already exist. The name of this account is **`UBrokerService`**. A default password provided for the account can be obtained by contacting Stonebranch support. We strongly suggest that you set this password at install time via the installation dialogs or via the **BROKERPWD** command line option.

Note

If the UBrokerService account already exists, its password cannot be changed from the install.

See Installing a Universal Agent System Install via the Graphical Interface for detailed information about setting up the Universal Broker service account at install time.

Using a Windows Domain Account to Execute Universal Broker

The Universal Broker Service may be configured to execute with an existing Windows domain account, provided that account has been granted the privileges listed in this section, above, on the local system and the account has the file system permissions described in Service Security.

Note

Depending on your environment, it may be necessary to add this account to the Domain Admins group. This will ensure the account has sufficient access to domain resources and is recognized as a true administrative account on all domain member systems that run the Universal Broker service as that account.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Agent components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with Manager Fault Tolerance enabled.
- Configuration information for Universal Agent components, when a local Universal Broker is operating in managed mode.

The default location for the spooled standard I/O files is **`.\Universal\spool\ucmdsrv`**.

The default location for the other database files is **`.\Universal\spool\ubroker`**.

Universal Command Server

Location

The spool directory must reside on a local device. It cannot reside on any network device, including network drives that may be mapped to a local drive. By default, the spool files are located in directory **`.\Universal\spool\ucmdsrv`**.

Space

You must have approximately 50 megabytes of disk space for the installation.

The amount of disk space required for the spool directory depend on the following factors:

- Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
- When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
- Average size of the user processes standard input, output, and error files. Keep in mind that spooling is not a feature for file transfer purposes. File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = required disk space.`

For example, if the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. This feature can be enabled through the Universal Configuration Manager.

For more information on the Manager Fault Tolerant feature, spooling of redirected standard I/O files, and Universal Configuration Manager, see the Universal Agent 7.2.x User Guide.

Security

Universal Broker and Universal Command Server require read/write access to the spool directory. No other Universal Agent components access the spool directory directly.

UAG Cache

UAG cache is used by Universal Automation Center Agent (UAG) for storing standard I/O files.

Space

Cache files are located, by default, in directory **`.\Universal\UagSrv\cache`**.

Cache files are created for each job that is run by Universal Agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by Universal Controller. You can configure the number of days that files remain in the cache using the Agent Cache Retention Period in Days Universal Controller system property.

The amount of disk space required for the cache directory depends on:

- Number of jobs you estimate will run during the cache retention period you specified.
- These files remain until they are purged by the automated cache purge process scheduled by Universal Controller daily at midnight.
- Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.`

For example:

If the files are purged every 7 days, and you run 1200 jobs on that agent server daily, and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

UAG automatically redirects the standard output and standard error files to the cache directory with no required user input.

Security

Universal Automation Center Agent (UAG) requires read/write access to the UAG cache directory. No other Universal Agent components access the cache directory. No general user access is required.

Universal Agent - User Mode Installation

Windows Versions

To install Universal Agent for Windows, you must have one of the following versions of Windows:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

Note

The Universal Agent for Windows User Mode installation package will execute only on a 64-bit version of Windows.

Additional Requirements

- Administrative authority to execute the installation.
- Network connectivity via TCP/IP.
- About 120 MB available disk space.
- Interactive access to the Windows system (command line support will be provided in an upcoming release).
- Microsoft Visual C++ 14 Runtime
- Microsoft Universal C Runtime

Note

The Universal Agent User Mode install does not provide the required Microsoft runtimes. If either of these do not exist, the install will issue an error and stop without installing any files. Both runtimes are available as redistributable packages from Microsoft. The Universal C Runtime is built into Windows 10, Server 2016, and Server 2019. It is available via a Microsoft update for Windows Server 2012 and Windows Server 20102 R2.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Installation Account

The account used to execute a Universal Agent for Windows user mode installation must have administrative privileges.

Universal Broker Service

The Universal Broker service runs either with an Administrative account or with the Local System account.

If you are using an Administrative account, the account must have the following privileges on the system upon which the Universal Agent resides (optional, but recommended privileges are noted):

- Act as part of the operating system
- Adjust memory quotas for a process
- Bypass traverse checking
- Debug programs
- Deny log on as batch job (optional, recommended)
- Deny log on locally (optional, recommended)
- Deny log on through Terminal Services (optional, recommended)
- Log on as a service
- Impersonate a client after authentication
- Increase scheduling priority
- Replace a process level token
- Take ownership of files and other objects

Selecting a Universal Broker Service Account

The Universal Agent for Windows user mode install will supply default credentials for an account named UBrokerService. This is the same default account that the Universal Agent for Windows system installation creates (if it does not already exist).

If the Universal Broker service will execute with a different account, you can provide that account's user ID and password during the install or specify them post-install in the Universal Broker service's configured settings within the Windows Services application.

If the Universal Broker service will execute with a domain account, that account must have the privileges listed above on the local Universal Agent system. The privileges can be granted locally or inherited via Active Directory policy, but they must be granted on the local system.

In all cases, the account chosen to run the Universal Broker service must be an existing Administrative account. The Universal Agent for Windows user mode install will not create it.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Agent components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with Manager Fault Tolerance enabled.
- Configuration information for Universal Agent components, when a local Universal Broker is operating in managed mode.

For user mode installations, the spool directory resides just under the main install directory.

- Most database files reside in `.\spool\ubroker`.
- Spooled standard I/O files reside in `.\spool\ucmdsrv`.

Universal Command Server

Location

The spool directory must reside on a local device. It cannot reside on any network device, including network drives that may be mapped to a local drive. By default, the spool files are located in directory `.\spool\ucmdsrv` (relative to the main installation directory).

Space

You must have approximately 125 MB available disk space for each instance of a Universal Agent for Windows user mode installation.

The amount of disk space required for the spool directory depend on the following factors:

- Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
- When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
- Average size of the user processes standard input, output, and error files. Keep in mind that spooling is not a feature for file transfer purposes. File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = required disk space.`

For example, if the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. This feature can be enabled by updating the Universal Command Server configuration file, `.\conf\ucmds.conf`.

For more information on the Manager Fault Tolerant feature and spooling of redirected standard I/O files, see the Universal Agent 7.2.x User Guide.

Security

Universal Broker and Universal Command Server require read/write access to the spool directory. No other Universal Agent components access the spool directory directly.

UAG Cache

The Universal Automation Center Agent (UAGSRV) stores standard I/O redirected from processes it executes to files in the UAG cache locations below.

Space

Cache files are located, by default, in directory `.\UagSrv\cache`, relative to the main Universal Agent for Windows user mode installation directory.

Cache files are created for each job that is run by Universal Agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by Universal Controller. You can configure the number of days that files remain in the cache using the Agent Cache Retention Period in Days Universal Controller system property.

For each instance of a Universal Agent user mode installation, the amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Controller daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.`

For example:

If the files are purged every 7 days, and you run 1200 jobs daily on a particular user mode Universal Agent and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

UAG automatically redirects the standard output and standard error files to the cache directory with no required user input.

Security

Universal Automation Center Agent (UAG) requires read/write access to the UAG cache directory. No other Universal Agent components access the cache directory. No general user access is required.

Universal Enterprise Controller (UEC)

Windows Versions

- Windows XP SP3
- Windows Server 2003 SP1 and higher
- Windows Vista
- Windows 7
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012

Additional Requirements

- An account with administrative privileges.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 35 megabytes of disk space.

Universal Enterprise Controller Client Applications

Windows Versions

- Windows XP SP3
- Windows Server 2003 SP1 and higher
- Windows Vista
- Windows 7
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012

Note

Itanium not supported for Windows Server 2003 and Windows Server 2008.

Additional Requirements

- For a per-machine install, Administrator access is required. For a per-user install, any account with the authority to install software can be used.
- The following conditions must be met before the UEC Client Applications can be successfully installed:
  - Account used for the installation must have write access to the desired destination folder.
  - No system policies (defined in Windows by your system administrator) may be in place that prohibit non-Administrative accounts from installing software.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 5 megabytes of disk space. This value does not include space required for the Java Runtime Environment (JRE). See the JRE installation requirements, available from Sun, for more information.
- Sun Java Runtime Environment (JRE 1.5 or above).

Java Runtime Environment

To run the UEC Client Applications, you also must have the Java Runtime Environment (JRE) version 1.5 or above. You can download the latest JRE installation package directly from Sun's website, www.java.com.

A Stonebranch Tip

Changes in Sun's versioning scheme for Java has created some confusion.

With version 1.2 of the Java environment, Sun began referring to Java as Java 2. The formal name was actually Java 2 with SDK 1.2.

As of January 2008, version 1.6 is latest version of the Java environment - a newer version than the original Java 2.

Universal Agent for SOA

Universal Agent for SOA on Windows is now available as 6.7.0.0. It is a 64-bit package, supported on the same Windows versions as the Agent: Windows 7, Server 2008 R2, Server 2012, Server 2012 R2, Server 2016, and Server 2019.

Note

Starting with the 6.7.0.0 release, the Universal Agent for SOA on Windows is a 64-bit application that replaces all previous versions. There is no upgrade path from the older 32-bit version to the 64-bit implementation. If the install detects an existing 32-bit version, it will prompt you to remove that version before proceeding.

You are responsible for preserving your existing configuration in this scenario. PLEASE remember to collect the license information stored in the `uacs.conf` file so that you can re-apply it following the 64-bit install. If you lose your license information, you will need to contact Stonebranch customer support to retrieve that information.

Additional Requirements

- An account with administrative privileges.
- Possible reboot. A reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1283 is installed, or required files are in use at the time of the installation.
- TCP/IP.
- About 110 megabytes of disk space.
- A supported 64-bit Universal Agent.

Installation Summary

Step 1	Download the distribution file from the Stonebranch Customer Portal.
Step 2	Log on to Windows using an account with the privileges noted above.
Step 3	Exit all running programs to reduce the likelihood of a reboot.
Step 4	Execute the installation file.
Step 5	Follow installation instructions presented by the graphical installation program.

Customization

Customization consists of starting the Universal Configuration Manager control panel applet and modifying the configuration options as required by the local environment.

Installation Requirements and Summary - UNIX

- Objective
- Universal Agent for UNIX
  - UNIX Versions
  - Additional Requirements
  - Platform Requirements
  - Directories and Files
- Universal Agent for SOA
  - UNIX Versions
  - Platform Requirements
  - Additional Requirements
- Installation Summary
  - Customization

Objective

This page provides installation requirements and summary for Universal Agent 7.2.x on a UNIX system.

Universal Agent for UNIX is comprised of one required package - Universal Agent (the Agent) - and one optional package: Universal Agent for SOA.

Universal Agent for UNIX

UNIX Versions

One of the following UNIX operating systems that are supported by Universal Agent 7.2.x:

- AIX 7.1 TL5 and above or AIX 7.2 TL2 and above
- HP-UX 11.23 and above (HP-UX IA64 package)
- Solaris 10 and above (SPARC-based)

Note

Universal Agent for Solaris 10 systems require libxnet.so version SUNW_1.3.1 or higher.

Note

The Python 3.7 Distribution for Universal Agent requires libintl.so version 0.18.1.1, REV=2011.03.15 or higher. The install will proceed if the required dependency does not exist, but the UA-specific python instance will not run until libintl is available.

- Solaris 10 and above (Intel-based)

Note

The Python 3.7 Distribution for Universal Agent requires libintl.so version 0.18.1.1, REV=2011.03.15 or higher. The install will proceed if the required dependency does not exist, but the UA-specific python instance will not run until libintl is available.

- Linux Environments with the following qualifications:
  - Intel x86_64-based system
    - Linux Kernel 3.10 or greater
    - glibc 2.17-78 or higher
  - IBM PowerPC-based system
    - Linux Kernel 3.10 or greater
    - glibc 2.17-157 or higher
    - Configured for little endian (LE)
  - zSeries 64-bit (S/390) system
    - Linux Kernel 2.6 or greater
    - glibc 2.4-31 or greater
  - RedHat Package Manager (RPM) distributions
  - Debian distributions
    - Linux Kernel 3.0 or greater
    - glibc 2.15 or higher

The list of supported systems grows rapidly. Contact Stonebranch, Inc. for a current list of supported UNIX operating systems if you require support for a system that is not listed.

Additional Requirements

- Superuser (root) access.
- TCP/IP Socket implementation.

- Approximately 400 megabytes of disk space for the installation. More disk space is required for variable files, such as log files, spool files, and trace files.
- Bourne shell or compatible.

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x to make sure that your platform is supported before performing an installation.

Directories and Files

All product files that are written to during product execution are stored in the **/var/opt/universal** directory by default. This section documents the estimated amount of space required on the file system for all directories (and their sub-directories), required security access, and mount requirements.

Command Reference Directory

Universal Command Server can execute commands of type **cmdref**. A command reference is a predefined command or script to which the Universal Command Manager refers by its file name.

The command reference directory is **/var/opt/universal/cmdref**.

Space

The amount of space required is solely dependent on the number of command reference files you define. No command reference files are included in the installation.

Security

Universal Command Server requires read access to the **cmdref** directory. The product administrator requires read/write access in order to maintain the command reference files. No general user access is required.

Log Directory

Universal Broker can be configured to write its messages to a log file.

The current log file and previous log file generations are stored in the **/var/opt/universal/log** directory.

Space

A log file size grows to about 150,000 bytes and is then rolled over to a generation file. Five generations of log files are saved. The oldest generation log file is deleted. The amount of space required for the five generations and the current log file is about 900,000 bytes.

Security

Universal Broker requires read/write access to the log directory and read/write access to all files in the log directory. No other Universal Agent components use the log directory at this time. No general user access is required.

Trace Directory

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. Customer Support to resolve product problems.

Space

Trace files can grow to significant size depending on how long the trace is active and how much work the program is doing during the tracing period. A trace file size of about 10MB is not unusual.

Trace file sizes can be limited by setting the maximum number of lines to write to a file with the MAX_TRACE_LINES configuration option. Once the maximum is reached, the trace file will wrap to the beginning.

Under normal operation, no space is required for trace files. Trace files are requested by Stonebranch, Inc. Customer Support only for problem resolution. When trace files are required, at least 20MB of disk space should be available.

Security

Universal Broker and the Broker components (Universal Command Server and Universal Control Server) require read/write access to the trace directory. No other Universal Agent components access the trace directory. No general user access is required.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Agent components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with Manager Fault Tolerance enabled.

The default location for the spooled standard I/O files and other database files is the **/var/opt/universal/spool** directory.

Universal Command Server

Space

Spool files are located, by default, in directory **/var/opt/universal/spool**. The location of various product files can be can be configured via the product configuration files.

The amount of disk space required for the spool directory depends on:

1. Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
   When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
2. Average size of the user processes standard input, output, and error files.

A Stonebranch Tip

Keep in mind that spooling is not a feature for file transfer purposes.

File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = required disk space.`

For example:

If the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. The feature must be turned on through the ALLOW_SPOOLING configuration settings.

For more information on the Manager Fault Tolerant feature and the spooling of redirected standard I/O files, see the Universal Agent 7.2.x User Guide.

Security

Universal Broker, Universal Command Server, and Universal Event Monitor require read/write access to the spool directory. No other Universal Agent components access the spool directory. No general user access is required.

Mount

The spool directory must be mounted on a local device. It cannot be mounted on a network device, such as an NFS or SAMBA mount.

UAG Cache

UAG cache is used by Universal Automation Center Agent (UAG) for storing standard I/O files.

Space

Cache files are located, by default, in directory **`/var/opt/universal/uag/cache`**.

Cache files are created for each job that is run by the Agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by Universal Controller. You can configure the number of days that files remain in the cache via the Universal Controller user interface (see the Agent Cache Retention Period in Days Universal Controller system property).

The amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Controller daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

`(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.`

For example:

If the files are purged every 7 days, and you run 1200 jobs on that agent server daily, and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

The Universal Automation Center Agent Server automatically redirects the standard output and standard error files to the cache directory with no required user input.

Security

Universal Automation Center Agent require read/write access to the UAG cache directory. No other Universal Agent components access the cache directory. No general user access is required.

Universal Agent for SOA

UNIX Versions

To install Universal Agent for SOA for UNIX, you must have a Linux environment that satisfies the following requirements:

- RedHat Package Manager (RPM) distributions
  - Intel x86_64-based system
  - Linux Kernel 3.10 or greater
  - gllibc 2.17-78 or higher
- Debian distributions
  - Linux Kernel 3.0 or greater
  - glibc 2.15 or higher

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x to make sure that your platform is supported before performing an installation.

Additional Requirements

- 512MB RAM minimum, 1 GB or more preferred.
- 150 MB free disk space.
- TCP/IP socket implementation.
- Superuser (root) access.
- Universal Agent 6.3.0.1 or later.

Installation Summary

Step 1	Download the distribution file from the Stonebranch Customer Portal.
Step 2	Copy the distribution file to the UNIX system.
Step 3	Logon to the UNIX system as root or execute **su** (substitute user) command to root.
Step 4	Uncompress the distribution file.
Step 5	Extract the files from the uncompressed **tar** file.
Step 6	Run the Stonebranch, Inc.-supplied script.

Customization

Customization consists of editing one or more of the product configuration files.

Installation Requirements and Summary - IBM i

- Objective
- System Requirements
- Platform Requirements
- Libraries and Logs
  - Universal Broker Job Log
  - Command Reference Library
  - Trace File Location Library
  - Spool Library
- Installation Summary
- Customization

Currently, IBM i runs Workload Automation 5.1.1. These pages provide information for that version.

Objective

This page provides installation requirements and summary for Workload Automation 5.1.1 on an IBM i system.

System Requirements

The requirements for installation of Workload Automation for IBM i are:

- IBM i V5R4M0 or above.
- TCP/IP.
- User profile with *ALLOBJ, *SPLCTL, *JOBCTL, and *SECADM authorities.
- About 360 megabytes of disk space.
- Network-attached workstation.
- PTF SI27629 installed on V5R4 systems.
  For additional information, see the documentation for APAR SE28859 and PTF SI27629 at both:
  - https://www\-912.ibm.com/n_dir/nas4apar.nsf/aaf5d88f9cc2ee10862571020058635c/a2b5a3b3ac874a9d862572d8003c7457?OpenDocument&Highlight=2,QGYOLJBL
  - https://www\-912.ibm.com/a_dir/as4ptf.nsf/c2fd98f5d2eccb83862574ce00520341/7529dd654f63941b862572eb00581099?OpenDocument

The Workload Automation 5 for IBM i installation process creates a product user profile, **UNVUBR511**, that is given ***ALLOBJ** special authority. **UNVUBR511**, with ***ALLOBJ** special authority, is required to successfully complete the installation.

Note

Some organizations and companies require the removal of ***ALLOBJ** authority from non-administrative user profiles. Workload Automation may be configured to run without ***ALLOBJ** authority; however, to do so requires additional administrative overhead. The steps required to remove product ***ALLOBJ** authority are described in Universal Broker Security.

The ***BASE** option contains the complete installation. This includes all of the components and utilities listed above.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Libraries and Logs

Under the IBM i native file system, Workload Automation write to product files residing in various libraries and write information to job logs.

This section specifies the following information for various libraries and logs:

- Estimated amount of space required
- Required security access
- Location of the production files

Universal Broker Job Log

Under IBM i, Universal Broker writes its messages to the **UBROKER** job log.

Space

Job log file growth is dependent on use of the Workload Automation Servers. The disposition and size of the job log depends on the job definition as well as system variables **QJOBMSGQMX** (maximum job log size) and **QJOBMSGQFL** (action when job log is full).

Security

Since Universal Broker uses a normal job log, no special security is required.

Command Reference Library

Universal Command Server can execute commands of type **cmdref**. A command reference is a predefined command or script to which the Universal Command Manager refers by its file name.

The default command reference library name is **UNVCMDREF**. For security reasons, the name of this library can be changed only via the Universal Command Server configuration, located in the **UNVCONF** file and **UCMDS** member. The configuration file entry has precedence over the default value. If the name is changed to something other than the default or configured value, Universal Command Manager will no longer be able to read the command reference files; it will generate appropriate error messages if an attempt is made to do so.

Space

The amount of space is dependent solely on the number of command reference files defined.

Security

Universal Command Server requires read access to the **UNVCMDREF** library files. Administrator accounts require appropriate access in order to maintain the command reference files. No general user access is required.

Trace File Location Library

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. Customer Support to resolve product problems.

On IBM i, Workload Automation trace files normally are written to ***CURLIB** (current library). Under Universal Broker and the associated servers, the current library is the temporary library designated at installation time (**UNVTMP511**).

Otherwise, current library is the default current library of the user.

Space

Trace files can grow to significant size, depending on how long the trace is active and how much work the program is doing during the tracing period. Due to the information associated with IBM i pointers and fact that physical database files are fixed record lengths, trace files quickly can grow very large on a system with high Universal Broker and server activity.

If the trace file size is increased beyond 500,000 records, the maximum file size must be changed or the associated job will hang. The hang results from a system generated inquiry message, which is issued when the maximum file size is exceeded. By default, trace files wrap before reaching the maximum file size, thus avoiding the system inquiry message.

The MAX_TRACE_LINES configuration option sets the number of records at which the trace file wrapping occurs. When the maximum size is reached, the trace file will wrap to the beginning.

No space is required under normal operation for trace files. Trace files are requested by Stonebranch, Inc. Customer Support only for problem resolution. When trace files are required, at least 40MB of disk space should be available.

Security

The **UNVUBR511** user profile requires at least ***CHANGE** authority to the **UNVTMP511** library to create and use the Universal Broker and server trace files. No general user access is required.

Spool Library

The spool library is used to store the following types of information:

- Execution information for Workload Automation components started by Universal Broker, **UBR_CMP_DB**.
- Universal Command Server status, **SRV_CMP_DB**.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with manager fault tolerance enabled.

Spool files are stored in the **UNVSPL511** library.

Space

The spool files are located in the **UNVSPL511** library. The amount of disk space required for the spool directory depends on these factors:

1. Number of spooling user processes that will be executing simultaneously.
   A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50. Once a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
2. Average number of records of the user processes for standard input, output, and error files.
   Each record is 32764 bytes in length. Keep in mind that spooling is not intended as a feature for file transfer purposes. File transfer related processes should execute without spooling enabled.

Once these numbers are determined, the average amount of disk space is calculated with the following formula:

`MAX-PROCESSES x AVERAGE NUMBER-OF-REC0RDS x 32764 = required disk space.`

As an example, if the maximum number simultaneous user processes is estimated at 20 and the average number of records in files is 1,000 bytes, the average amount of required disk space is 655MB (20 x 1000 x 32764).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. The feature must be turned on through the ALLOW_SPOOLING configuration settings.

Further details on the Manager Fault Tolerant feature and spooling of redirected standard I/O files can be found in the Universal Agent 7.2.x User Guide.

Security

The **UNVUBR511** user profile requires at least ***CHANGE** authority to the **UNVSPL511** library to create and use the spool files. No general user access is required.

No other Workload Automation 5 for IBM i components access the spool library.

Installation Summary

Warning

This installation procedure is intended only for default installations into simple environments. Installing into high availability environments may require a customized installation. For customized installation, parallel installation or multiple system roll out see the Installation Guide for IBM i.

Step 1	Download the distribution file from the Stonebranch Customer Portal to a computer running a supported version of Windows or UNIX.
Step 2	Extract the Licensed Program Save File from the distribution file.
Step 3	If transferring the installation file to the native, QSYS file system, create a save file, UNV511, on the target IBM I system in library QGPL. This step is not necessary if transferring the file to the IFS.
Step 4	FTP the Licensed Program Save File to either the save file created in step 1 or to /QSYS.LIB/QGPL.LIB/UNV511.SAVF.

Step 5	Execute the following command to install Workload Automation 5 for IBM i to the default libraries:
	`RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511)`
	Add the **FRCOBJCVN** parameter when installing on a version of IBMi that is greater than the version identified in the distribution file. For example, when using the 7.2-targeted distribution file to install on IBMi 7.4, enter:
	`RSTLICPGM LICPGM(0UNV511) DEV(*SAVF) SAVF(QGPL/UNV511) FRCOBJCVN(*YES *ALL)`

Step 6	Verify the correctness of the install by executing the following command: `CHKPRDOPT OUNV5120`
Step 7	Optionally, rename the UNVUBR511 subsystem to UBROKER for consistent operation across future installations.

Customization

Customization consists of editing configuration members in the UNVPRD511/UNVCONF source file.

Universal Agent 7.2.x Installation Quick Start Guides

The following Quick Start Guides are provided for Universal Agent 7.2.x installation:

- Universal Agent 7.2.x for z/OS Quick Start Guide
- Universal Agent 7.2.x for Windows Quick Start Guide
- Universal Agent 7.2.x for UNIX Quick Start Guide

Universal Agent 7.2.x for z/OS Quick Start Guide

- Objective
- System Requirements
- Platform Requirements
- Downloading the Universal Agent 7.2.x for z/OS Distribution File
- Installing Universal Agent 7.2.x for z/OS
- Licensing your Universal Agent 7.2.x for z/OS Components
  - Universal Command Manager License File
  - Universal Data Mover Manager License File
  - UNVCONF Product Library
  - Restart Universal Broker
- Encrypting User ID and Password for Job Execution
- Executing a Universal Command Job
- Executing a Universal Data Mover Job
- Running a System Query

Objective

This Universal Agent 7.2.x for z/OS Quick Start Guide provides a short summary on how to install Universal Agent on a z/OS system and run a few simple batch jobs demonstrating Universal Agent on z/OS. For a complete description of the installation process, see Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance.

The following information is provided:

- System Requirements
- Downloading the Universal Agent 7.2.x for z/OS distribution file.
- Installing Universal Agent 7.2.x on z/OS.
- Licensing Universal Agent components.
- Encrypting passwords to be used in Universal Agent jobs.
- Executing a Universal Command job.
- Executing a Universal Data Mover job.
- Executing a Universal Broker query job.

System Requirements

- z/OS 2.2, 2.3, or 2.4.
- SMP/E 3.5 or later.
- IBM Communication Server for z/OS 2.2, 2.3, or 2.4.
- IBM Language Environment (LE) for z/OS 2.2, 2.3, or 2.4.
- Windows workstation capable of establishing a TCP/IP network connection to the z/OS system.
- TSO user ID with an OMVS segment.
- About 1900 cylinders of DASD.
- Two available TCP/IP ports on z/OS.

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x to make sure that your platform is supported before performing an installation.

Downloading the Universal Agent 7.2.x for z/OS Distribution File

The Universal Agent 7.2.x for z/OS product distribution file - **`sb-7.2.x.x-zos.tar.Z`** - is available for download from the Stonebranch Customer Portal.

A customer user name and password - provided by Stonebranch, Inc. - are required to access the Customer Portal.

Installing Universal Agent 7.2.x for z/OS

Step 1	Download one of the Universal Agent 7.2.x for z/OS product distribution files.
Step 2a	If one of the distribution files with the .tar.Z extension was downloaded, decompress it using any utility that can process files in Lempel-Ziv (that is, **compress**) format. Use **tar** to extract the files from the uncompressed file. To do both in a single statement, decompress the file to stdout and pipe the result to **tar** (for example, **zcat sb-v.r.m.l-zos.tar.Z | tar xvf -)**.

Step 2b	If one of the distribution files with the .zip extension was downloaded, decompress it using any utility that can process files in **ZIP** (or, PKZIP) format. The .zip file can also be extracted using Windows native file extraction tools.
Step 3	Modify the UNVFTP.TXT script provided by the distribution file. This script contains statements to upload the installation files and sample JCL, which can be used to process the files after upload. Two versions of these files are included in the distribution file. The files in the main directory are EBCDIC-encoded files. ASCII-encoded versions of these files are in the `./ascii` subdirectory. See z/OS Installation - Transferring Installation Files to z/OS for complete details.
Step 4	Modify the JCL uploaded from UNVRECV.JCL and run the job to receive the TSO XMIT file(s). If the distribution file that contains the DFDSS archive is used, UNVRECV.JCL also contains a step to restore the contents of the archive. See Receive the Installation Datasets for complete details.
Step 5	Edit the INSTALL(#SETUP) member to meet local requirements as described in the #SETUP comment prolog. #SETUP creates customized installation JCL members in the INSTALL library. Execute the installation jobs as required for your SMP/E environment. See z/OS Installation - SMPE Installation for complete details.
Step 6	Create started task user and group profiles for Universal Broker and Universal Enterprise Controller. See z/OS Configuration - Started Tasks for complete details.
Step 7	Define the SUNVLOAD library to APF, Program Control, and LNKLST. See z/OS Configuration - Load Library for complete details.
Step 8	Define SMF exits for Universal Command and Universal Automation Center Agent. See z/OS Configuration - SMF Exits for complete details.
Step 9	Configure the UNIX System Services file systems used by the Universal Broker and Universal Enterprise Controller. See Universal Agent Database Configuration for complete details.
Step 10	Perform individual component customization, including JCL procedure edits and adding component licenses. See z/OS Installation - Customization for complete details.
Step 11	Follow the optional steps described in z/OS USS Installation to complete the installation, if desired.

Licensing your Universal Agent 7.2.x for z/OS Components

Note

A Universal Agent eventually will be required to obtain its license information from a Universal Controller or UCAM installation. Support for locally-configured license information will be removed at that time.

After Universal Agent 7.2.x for z/OS has been installed, you must configure the following Universal Agent components with product licenses before they can be used:

- Universal Command Manager
- Universal Connector
- Universal Data Mover Manager

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

The format of a product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: **Agent_Stonebranch_MVS_A1.txt**.

Universal Command Manager License File

For Universal Command Manager, **Agent** is used as the <*component name*> in the product license file name and as the name of the product in the file itself - see below.)

The following is a sample Universal Command Manager for z/OS product license file:

```
License_Product "AGENT"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2030.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Universal Data Mover Manager License File

For Universal Data Mover Manager, **MFT** is used as the *<component name>* in the product license file name and as the name of the product in the file itself - see below.)

The following is a sample Universal Data Mover Manager for z/OS product license file:

```
License_Product "MFT"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2030.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

UNVCONF Product Library

In the **UNVCONF** product library:

- Enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UCMCFG00**.
- Enter the Universal Connector license parameters into the Universal Connector configuration file, member **USPCFG00**.
- Enter the Universal Data Mover Manager license parameters into the Universal Data Mover Manager configuration file, member **UDMCFG00**.

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it.

Stop Universal Broker	`STOP UBROKER`
Start Universal Broker	`START UBROKER[,UPARM='options']`

Encrypting User ID and Password for Job Execution

Universal Command and Universal Data Mover typically require user credentials to execute work on a remote host. The user credentials can be stored in an encrypted format for privacy reasons. Read access to the file should be restricted to authorized users even though the contents are encrypted. Universal Command and Universal Data Mover will read user credentials in the encrypted format.

The following JCL example executes the UENCRYPT utility that encrypts command options it reads from the UNVIN ddname and writes them to the UNVOUT ddname in the form of an encrypted command file. This example creates member DALLAS in the PDS/E **`PROD.UNV.USERS`**. The DALLAS member will contain the user ID and password options specified in the UNVIN ddname in a UENCRYPT format.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1    EXEC PGM=UENCRYPT
//STEPLIB  DD  DISP=SHR,DSN=#SHLQ.UNV.SUNVLOAD
//UNVIN    DD  *
 -USERID USR100 -PWD ABCDEF
/*
//UNVOUT   DD  DSN=PROD.UNV.USERS(DALLAS),DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  DUMMY
```

Executing a Universal Command Job

The following JCL example executes the UNIX command "ls -l" on remote host DALLAS using the encrypted command file created above.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//         JCLLIB ORDER=#SHLQ.UNV.SUNVSAMP
//*
//STEP1    EXEC UCMDPRC
//USER     DD  DSN=PROD.UNV.USERS(DALLAS),DISP=SHR
//SYSIN    DD  *
-cmd 'ls -l' -host dallas -encryptedfile user
/*
```

Executing a Universal Data Mover Job

Universal Data Mover execution is driven by an easy to use scripting language. The Universal Data Mover script controls the transfer sessions established with remote hosts and the file and transfer operations performed on the host's file systems.

The following JCL example executes Universal Data Mover. Universal Data Mover always reads the script from the UNVSCR ddname. In this example, the script establishes a session with remote host DALLAS and transfers the data set allocated to ddname DATA to file app.data on the remote host. The default transfer mode of binary is used for transferring the data.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1    EXEC UDMPRC
//DATA     DD  DSN=APP.DATA,DISP=SHR
//USER     DD  DSN=PROD.UNV.USERS(DALLAS),DISP=SHR
//UNVSCR   DD  *
set echo=yes halton=warn
open dest=dallas xfile=user
filesys local=dd
copy local=data dest=app.data
quit
/*
```

Running a System Query

Use the Universal Query utility to query the Universal Broker for Broker and active component information. Universal Query returns information about the Universal Broker, as specified by configuration options, as well as information regarding the components managed by Universal Broker.

The following JCL example executes the UQUERY utility. In this example, UQUERY will query the Broker running on host DALLAS.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//         JCLLIB ORDER=#SHLQ.UNV.SUNVSAMP
//*
//STEP1    EXEC UQRYPRC
//SYSIN    DD  *
-host dallas
/*
```

Universal Agent 7.2.x for Windows Quick Start Guide

- Objective
- System Requirements
  - Windows Versions
  - Additional Requirements
  - Platform Requirements
- Downloading the Universal Agent 7.2.x for Windows Distribution File
- Installing Universal Agent 7.2.x via the Windows Installer Graphical Interface
- Licensing your Universal Agent 7.2.x for Windows Components
  - Universal Command Manager License File
  - Universal Data Mover Manager License File
  - Licensing Methods
  - Licensing via Universal Configuration Manager
  - Restart Universal Broker
- Encrypting User ID and Password for Job Execution
- Executing a Universal Command Job
- Executing a Universal Data Mover Job
- Running a System Query

Objective

This Universal Agent 7.2.x for Windows Quick Start Guide provides a short summary on how to perform a Universal Agent installation on a Windows system and execute Universal Command and Universal Data Mover jobs. For a complete description of the installation process, see Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance.

The following information is provided:

- System requirements
- Downloading the Universal Agent 7.2.x for Windows distribution file.
- Extracting and installing Universal Agent 7.2.x on Windows.
- Licensing Universal Agent components.
- Encrypting passwords.
- Executing a Universal Command job.
- Executing a Universal Data Mover job.
- Running a system query.

System Requirements

Windows Versions

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

System Installation	• The Universal Agent for Windows System Install package requires a 64-bit edition of the Windows versions listed above.
User Mode Installation	• The Universal Agent for Windows User Mode Install package will execute only on a 64-bit version of Windows.

Additional Requirements

System Install	• An account with administrative privileges. • Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation. • TCP/IP. • About 325 megabytes of disk space.

User Mode Install	• Administrative authority to execute the installation. • Network connectivity via TCP/IP. • About 120 MB available disk space. • Interactive access to the Windows system (command line support will be provided in an upcoming release). • Microsoft Visual C++ 14 Runtime • Microsoft Universal C Runtime Note The Universal Agent User Mode install does not provide the required Microsoft runtimes. If either of these do not exist, the install will issue an error and stop without installing any files. Both runtimes are available as redistributable packages from Microsoft. The Universal C Runtime is built into Windows 10, Server 2016, and Server 2019. It is available via a Microsoft update for Windows Server 2012 and Windows Server 20102 R2.

Platform Requirements

Since platform requirements may change with new releases of a product, please consult the Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x page to make sure that your platform is supported before performing an installation.

Downloading the Universal Agent 7.2.x for Windows Distribution File

The Universal Agent 7.2.x for Windows product distribution files (for System Install and User Mode Install) are available for download from the Stonebranch Customer Portal.

A customer user name and password - provided by Stonebranch, Inc. - are required to access the Customer Portal.

Installing Universal Agent 7.2.x via the Windows Installer Graphical Interface

Step 1	Download the appropriate Universal Agent 7.2.x for Windows product distribution file to your work station: **System Install** • **`sb-7.2.x.-windows-x64.exe`**, the Universal Agent for Windows System Install distribution file (for supported 64-bit versions of Windows only). **User Mode Install** • **`sb-7.2.MOD.LEVEL-windows-x64-um.exe`**, the Universal Agent for Windows User Mode Install distribution file (for supported 64-bit versions of Windows only).
Step 2	Execute the distribution file to extract the files. The installation starts after the files are extracted. It first will verify that your machine meets the minimum system requirements. If the requirements are met, a Welcome dialog displays.
Step 3	Click the <Next> button. A list of Universal Agent components included in the package then displays. A drive icon displays next to each item in the list, indicating that the component will be installed. The Install To: field identifies the location where the components will be installed. • If you want to select a different location, click the <Change...> button. • If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the <Space> button. • If you do not want to install a component: 1. Click the drive icon next to that component name. 2. From the drop-down list that displays, select the X icon to mark the component as one not to be installed.
Step 4	When you have selected the components (and their installation destinations) that you want to install, click the <Next> button to continue the installation process. Depending on the components selected, the installation may prompt for additional values, such as working folders for Server components. Follow the directions provided with each dialog to complete the installation.

Licensing your Universal Agent 7.2.x for Windows Components

Note

A Universal Agent eventually will be required to obtain its license information from a Universal Controller or UCAM installation. Support for locally-configured license information will be removed at that time.

After Universal Agent 7.2.x for Windows has been installed, you must configure the following Universal Agent 7.2.x components with product licenses before they can be used:

- Universal Command Manager
- Universal Connector for SAP
- Universal Connector for PeopleSoft

- Universal Data Mover Manager
- Universal Application Container Server

Note

Universal Enterprise Controller 6.3.x for Windows and Universal Application Container Server (as a component of Universal Command Agent for SOA 6.7.x for Windows) are packaged, and licensed, separately.

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: **Agent_Stonebranch_NT_A1.txt**.

Universal Command Manager License File

For Universal Command Manager, **Agent** is used as the <*component name*> in the product license file name and as the name of the product in the product license file itself - see below.)

The following is a sample Universal Command Manager for Windows product license file:

```
License_Product "AGENT"
License_Customer "STONEBRANCH"
License_OS_Type "NT"
License_Type "PERPETUAL"
License_Expiration_Date 2030.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Universal Data Mover Manager License File

For Universal Data Mover Manager, **MFT** is used as the <*component name*> in the product license file name and as the name of the product in the product license file itself - see below.)

The following is a sample Universal Data Mover Manager for Windows product license file:

```
License_Product "MFT"
License_Customer "STONEBRANCH"
License_OS_Type "NT"
License_Type "PERPETUAL"
License_Expiration_Date 2030.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Licensing Methods

Universal Agent 7.2.x for Windows components can be configured with product licenses either by:

1. Entering the information into their configuration file. It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)
    - Universal Command Manager: **ucmd.conf**
    - Universal Connector for SAP: **usap.conf**
    - Universal Connector for PeopleSoft: **upps.conf**
    - Universal Data Mover Manager: **udm.conf**
    - Universal Application Container Server: **uacs.conf**
2. Specifying the information on the Universal Configuration Manager License Information page for that component, either by.
    a. Entering the information specified in the license file.
    b. Importing the license file.

Note

Universal Configuration Manager is installed during the Universal Agent 7.2.x for Windows installation.

Licensing via Universal Configuration Manager

To enter license information via the Universal Configuration Manager, perform the following steps:

Step 1	On the Windows Control Panel, double-click the Universal Configuration Manager icon to display the Universal Configuration Manager screen.
Step 2	In the Installed Components tree, click **+** next to the component that you want to configure with license information.
Step 3	Click License Information to display the License Information page.
Step 4	If you want to enter the license information on the page, make sure that you enter the information exactly as it is in the license information file. All information is case-sensitive; punctuation and spacing must be maintained.
Step 5	If you want to import the license file: 1. Click the Import button. 2. On the Select License File dialog that displays, select the license file to import. 3. Click the Open button. The information is imported to the License information page.
Step 6	Click the OK button to save the license information.

Restart Universal Broker

For Universal Broker to read the license information, you must start / restart it.

Via Windows Services

Step 1	Go to Windows Services.
Step 2	Locate and select Universal Broker.
Step 3	If Universal Broker is not running, click **Start**. If Universal Broker is running, click **Restart**.

Via Universal Configuration Manager

If you enter license information via the Universal Configuration Manager (see above), Universal Broker is restarted automatically, and the license information is read, when you click OK to save the license information.

Encrypting User ID and Password for Job Execution

Job execution demands a user ID and password. This information should be stored in an encrypted file, which Universal Command and Universal Data Mover will process before starting a job.

Step 1	Create a file containing a user ID (-userid *id*) and a password (-pwd *password*) for each component.
Step 2	Run the Universal Encrypt utility to encrypt the contents of the file.
Step 3	Execute Universal Command and Universal Data Mover using the encrypted file (-encryptedfile *filename*) option.

Executing a Universal Command Job

The command line syntax for executing Universal Command for Windows is:

```
ucmd -cmd command | -script file -host hostlist -encryptedfile filename [additional options]
```

Executing a Universal Data Mover Job

The command line syntax for executing Universal Data Mover for Windows is:

```
udm
open [primary={*| local | host-name} [optional parameters]] secondary=host-name [optional parameters]
UDM command {required parameters} [optional parameters]
```

Running a System Query

Use the Universal Query utility to query the Universal Broker for Broker-related and active component-related information. Universal Query returns information for the Universal Broker, as specified by configuration options on the command line or in a configuration file, as well as information regarding the components managed by Universal Broker.

The command line syntax for executing Universal Query for Windows is:

```
uquery -host hostaddress [additional options]
```

Universal Agent 7.2.x for UNIX Quick Start Guide

- Objective
- System Requirements
  - UNIX Versions
  - Additional Requirements
  - Platform Requirements
- Downloading the Universal Agent 7.2.x for UNIX Distribution File
- Installing Universal Agent 7.2.x for UNIX
- Licensing your Universal Agent 7.2.x for UNIX Components
  - Universal Command Manager License File
  - Universal Data Mover Manager License File
  - Licensing Method
  - Restart Universal Broker
- Encrypting User ID and Password for Job Execution
- Executing a Universal Command Job
- Executing a Universal Data Mover Job
- Running a System Query

Objective

This Universal Agent 7.2.x for UNIX Quick Start Guide provides a short summary on how to install Universal Agent on a UNIX system and execute Universal Command and Universal Data Mover jobs. For a complete description of the installation process, see Universal Agent 7.2.x Installation, Upgrade, and Applying Maintenance.

The following information is provided:

- System requirements
- Downloading the Universal Agent 7.2.x for UNIX distribution file.
- Extracting and installing the Universal Agent 7.2.x on UNIX.
- Licensing Universal Agent components.
- Encrypting passwords.
- Executing a Universal Command job.
- Executing a Universal Data Mover job.
- Running a system query.

System Requirements

UNIX Versions

One of the following UNIX operating systems that are supported by Universal Agent 7.2.x:

- AIX 6.1 TL1 and above, 64-bit
- HP-UX 11.23 and above (HP-UX IA64 package)
- Solaris 10 and above (SPARC-based)

Note

Universal Agent for Solaris 10 systems require libxnet.so version SUNW_1.3.1 or higher.

Note

The Python 3.7 Distribution for Universal Agent requires libintl.so version 0.18.1.1, REV=2011.03.15 or higher. The install will proceed if the required dependency does not exist, but the UA-specific python instance will not run until libintl is available.

- Solaris 10 and above (Intel-based)

Note

The Python 3.7 Distribution for Universal Agent requires libintl.so version 0.18.1.1, REV=2011.03.15 or higher. The install will proceed if the required dependency does not exist, but the UA-specific python instance will not run until libintl is available.

- Linux Environments with the following qualifications:
  - Intel x86_64-based system
    - Linux Kernel 3.10 or greater
    - glibc 2.17-78 or higher
  - IBM PowerPC-based system
    - Linux Kernel 3.10 or greater
    - glibc 2.17-157 or higher
    - Configured for little endian (LE)
  - zSeries 64-bit (S/390) system

    - Linux Kernel 2.6 or greater
    - glibc 2.4-31 or greater
- RedHat Package Manager (RPM) distributions
- Debian distributions
    - Linux Kernel 3.0 or greater
    - glibc 2.15 or higher

The list of supported systems grows rapidly. Contact Stonebranch, Inc. for a current list of supported UNIX operating systems if you require support for a system that is not listed.

Additional Requirements

- Superuser (root) access.
- TCP/IP Socket implementation.
- Approximately 400 megabytes of disk space for the installation. More disk space is required for variable files, such as log files, spool files, and trace files.
- Bourne shell or compatible.

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 7.2.x and Universal Agent 7.2.x to make sure that your platform is supported before performing an installation.

Downloading the Universal Agent 7.2.x for UNIX Distribution File

Stonebranch Inc. provides separate Universal Agent 7.2.x for UNIX product distribution files for different UNIX operating systems:

- AIX
- HP-UX
- Solaris: Intel-Based and SPARC-Based
- Linux: x86_64-Based, IBM S/390, and zSeries

These product distribution files are available for download from the Stonebranch Customer Portal.

A customer user name and password - provided by Stonebranch, Inc. - are required to access the Customer Portal.

Installing Universal Agent 7.2.x for UNIX

Step 1	Download the Universal Agent 7.2.x for UNIX product distribution file to your work station.
Step 2	Issue a zcat command to uncompress and extract the installation files from the distribution file.
Step 3	To start the installation, issue the following command: **`sh ./unvinst`**
Step 4	The output from **unvinst** is written to file **install.log** in the current directory.

Licensing your Universal Agent 7.2.x for UNIX Components

Note

A Universal Agent eventually will be required to obtain its license information from a Universal Controller or UCAM installation. Support for locally-configured license information will be removed at that time.

After Universal Agent 7.2.x for UNIX has been installed, you must configure the following Universal Agent 7.2.x components with product licenses before they can be used:

- Universal Command Manager
- Universal Connector for SAP
- Universal Connector for PeopleSoft
- Universal Data Mover
- Universal Application Container Server

Note

Universal Application Container Server (as a component of Universal Command Agent for SOA 6.7.x for UNIX) is packaged, and licensed, separately.

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

The format of the product license file name is: <*component name*>_<*customer name*>_<*operating system*>_<*schedule or solution*>.txt. For example: **Agent_Stonebranch_UNIX_A1.txt**.

Universal Command Manager License File

For Universal Command Manager, **Agent** is used as the <*component name*> in the product license file name and as the name of the product in the product license file itself - see below.

The following is a sample Universal Command Manager for UNIX product license file:

```
License_Product "AGENT"
License_Customer "STONEBRANCH"
License_OS_Type "UNIX"
License_Type "PERPETUAL"
License_Expiration_Date 2030.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Universal Data Mover Manager License File

For Universal Data Mover Manager, **MFT** is used as the <*component name*> in the product license file name and as the name of the product in the product license file itself - see below.

The following is a sample Universal Data Mover Manager for UNIX product license file:

```
License_Product "MFT"
License_Customer "STONEBRANCH"
License_OS_Type "UNIX"
License_Type "PERPETUAL"
License_Expiration_Date 2030.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Licensing Method

Enter each component's product license file information into its configuration file:

- Universal Command Manager: **ucmd.conf**
- Universal Connector for SAP: **usap.conf**
- Universal Connector for PeopleSoft: **upps.conf**
- Universal Data Mover Manager: **udm.conf**
- Universal Application Container Server: **uacs.conf**

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it:

Stop Universal Broker	`ubrokerd stop`

Start Universal Broker	`ubrokerd start`

Encrypting User ID and Password for Job Execution

Job execution demands a user ID and password. This information should be stored in an encrypted file, which Universal Command and Universal Data Mover will process before starting a job.

Step 1	Create a file containing a user ID (-userid *id*) and a password (-pwd *password*).
Step 2	Run the Universal Encrypt utility to encrypt the contents of the file.
Step 3	Execute Universal Command and Universal Data Mover using the encrypted file (-encryptedfile *filename*) option.

Executing a Universal Command Job

The command line syntax for executing Universal Command for UNIX is:

```
ucmd -cmd command | script file -host hostlist -encryptedfile filename [additional options]
```

Executing a Universal Data Mover Job

The command line syntax for executing Universal Data Mover for Windows is:

```
udm
open [primary={*| local | host-name} [optional parameters]] secondary=host-name [optional parameters]
UDM command {required parameters} [optional parameters]
```

Running a System Query

Use the Universal Query utility to query the Universal Broker for Broker-related and active component-related information. Universal Query returns information for the Universal Broker, as specified by configuration options on the command line or in a configuration file, as well as information regarding the components managed by Universal Broker.

The command line syntax for executing Universal Query for UNIX is:

```
uquery -host hostaddress [additional options]
```